00188.46-01

EDA Build-1
Final Report

Susan Dorsky
Susan Lockwood
Mark Peters
George Couluris

December 2000

Prepared for:
NASA Ames Research Center
Moffett Field, CA
Contract No. NAS2-98005

Prepared by:

Seagull Technology, Inc.
16400 Lark Avenue
Los Gatos, CA 95032-2547

Acknowledgment

Seagull Technology Inc., as a subcontract to Systems Resources Corporation (SRC), developed the Miles-In-Trail Spacing Tool for EDA software. The authors appreciate the extensive discussions, technical direction and program management of Steve Green and Rich Coppenbarger within the En route Operations branch of NASA Ames Research Center. The authors also wish to thank the entire CTAS software group for their help with understanding the CTAS code and group procedures. We would especially like to recognize Karen Tung, Akbar Sultan, Michael Grace, Darrell Wooten, Dr. Danny Chiu and Philippe Stassart for their assistance and encouragement.

Table of Contents

1 Introduction

1.1 Purpose

This document details the results of the RTO 46 Build-1 design and implementation as stated in Reference 1. Build 1 has the objectives to add to the CTAS En Route Descent Advisor a *Miles-In-Trail Spacing Tool* and to integrate that tool with the existing Conflict Detection and Trial Planning features. This document is intended to serve as a reference to NASA/EDA researchers. It supersedes Ref. 6 as it includes updated material from the Build-1 design document along with items that resulted from the presentation of the design to the customer and CTAS software personnel.

1.2 Scope

This scope of this document encompasses Build 1 of the En route Descent Advisor (EDA), which is planned as a series of five builds. Builds 1 and 2 will be research tools that support near-term part-task assessments of trajectory-based operations without regard to sector and facility jurisdictional structures and associated inter-sector coordination. These initial builds will introduce and examine CD&R and flow conformance integration.

- Build 1 will focus on en route spacing.
- Build 2 will focus on arrival metering.
- Build 3 will be a new and methodical rendering of EDA to provide an experimental platform for high-fidelity controller-in-the-loop simulations of real-world, operational environments.
- Build 4 will further enhance EDA, incorporating advanced communication, navigation and surveillance (CNS) capabilities.
- Build 5 will evolve EDA for final technical transfer and demonstration.

Build 1 of the EDA adds a single new feature to the CTAS system: a Miles-In-Trail (MIT) Spacing Tool. This tool has three components:

1. An interface for definition of Spacing References (Section 3),
2. An interface for the selection of a Spacing Stream (Section 4), and
3. A means to identify the aircraft approaching that Spacing Reference, called a Spacing Stream (Section 6).

Build 1 will also integrate the new MIT tool with the existing Conflict Detection and Trial Planning tools (Section 7).

1.3 Acronyms

AATT	Advanced Air Transportation Technologies
ATM	Air Traffic Management
CD&R	Conflict Detection and Resolution
CNS	Communications, Navigation, and Surveillance
CPTP	Conflict Probe and Trial Planning
CTAS	Center TRACON Automation System
D2	Direct To

DAG-TM	Distributed Air-Ground Traffic Management
DDTS	Distributed Defect Tracking Solution
DST	Decision Support Tool
EDA	En Route Descent Advisor
FMS	Flight Management System
GUI	Graphical User Interface
IFR	Instrument Flight Rules
MIT	Miles-In-Trail
PFS-C	CTAS Profile Selector - Center
PGUI	Planview Graphical User Interface
PVD	Planview Display
TFM	Traffic Flow Manager
TGUI	Timeline GUI
TRACON	Terminal Radar Approach Control
TS	Trajectory Synthesizer

1.4 Terms

Free flight	A safe and efficient flight operating capability under IFR in which operators have the freedom to select their path and speed in real time (from Ref 2).
Spacing Reference	A point, line, or arc which is used as a reference to assist in the spacing of aircraft by a controller.
Spacing List Window	A window opened on the CTAS PGUI workstation showing the spacing stream of aircraft passing over/near a spacing reference.
Modify Spacing Reference Window	A window opened on the CTAS PGUI permitting the controller to define the characteristics of a spacing reference.
Aircraft Selection Filter	A window opened on the CTAS PGUI permitting the controller to define the automatic selection criteria for including aircraft into a spacing stream.

1.5 References

1. NRA Task Order 46, En route Descent Advisor (EDA), Build-1 Design and Development, and Build-2 Initial Design", NASA Ames.
2. RTCA Taskforce 3, Select Committee on Free Flight, Jan 1995
3. Green, S., Grace, M., "Conflict-free Planning for En route Spacing: A Concept for Integrating Conflict Probe and Miles-In-Trail", AIAA-99-3988, AIAA Guidance, Navigation and Control Conference, Portland OR 1999.
4. Green, S.M, "En route Spacing Tool: Efficient Conflict-Free Spacing to Flow-Restricted Airspace" 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, June 2000
5. McNally D., Erzberger, H., Bach R., Chan W., "A Controller Tool for Transition Airspace". AIAA-99-3988, AIAA Guidance, Navigation and Control Conference, Portland OR 1999.

6. Dorsky S., et al., "EDA Build-1 Design Document", Seagull Technology Draft Report, September 2000.

1.6 Requirements Cross Reference

Appendix A found at the end of this document is a requirements cross-reference between Reference 1, RTO 46 EDA Build 1, and this document. The purpose of this is to ensure that all Build 1 intended tasks are being addressed by this document. We have gone through Reference 1 and extracted all stated requirements related to Build-1. These have been assigned an internal Seagull number (Column 1). The text of the requirements is contained in Column 2. The section of this document addressing the requirement is located in Column 3. If the requirement is not specifically applicable to Build 1 and/or there is no specific action required, Column 3 contains N/A (not applicable).

2 Miles-In-Trail Spacing Tool, An Overview

2.1 Problem Definition

The purpose of the EDA Miles-In-Trail Spacing tool is to give the controller assistance in positioning a selected set of aircraft in relationship to a spacing reference defined by that controller or by the CMU. The application of this tool is varied and could be used to meter aircraft into a local arrival fix, or, alternatively, could be used to establish en route spacing for some distant airport or fix.

The Miles-In-Trail tool handles three different types of spacing references: Points, Lines, and Arcs. Section 3 describes the methods for defining the characteristics of a spacing reference. Future reference types may include those comprised of polygonal boundaries as well as sets of unrelated reference objects.

The Build 1 Miles-In-Trail Spacing Tool is offered as a research platform to test the metering concept and provide experience for subsequent operational versions. The Build 1 capability supports a single reference and a single stream with manually selected aircraft. However, the design in no way precludes provisions for the definition of multiple references, the automatic selection of aircraft into a Spacing Stream, and the simultaneous support of multiple spacing streams for multiple spacing references. These enhancements are planned for future versions of the tool.

2.2 Solution Overview

Making maximum use of existing functionality in the CTAS baseline, and adding new functionality as necessary, Seagull's engineers have designed a compact and easy to use tool for spacing aircraft to a reference. A wealth of information about the Spacing Stream is displayed in the tool's Spacing List window: position of the lead aircraft and order of trailing aircraft, time of fix crossing, distance to the reference, delta distance from desired spacing (positive or negative), trial plan distance, whether or not the selected flight will actually cross the reference, and an indication of whether this flight falls "out of bounds" of some pre-configured tolerance.

The user starts the tool, defines a Spacing Reference, and selects aircraft for inclusion in the Spacing Stream. On the Planview GUI (PGUI), a constantly updated window reflects information about the Stream as detailed in the previous paragraph (see Figure 2-1 for a graphical representation). The list is dynamically updated in real time.

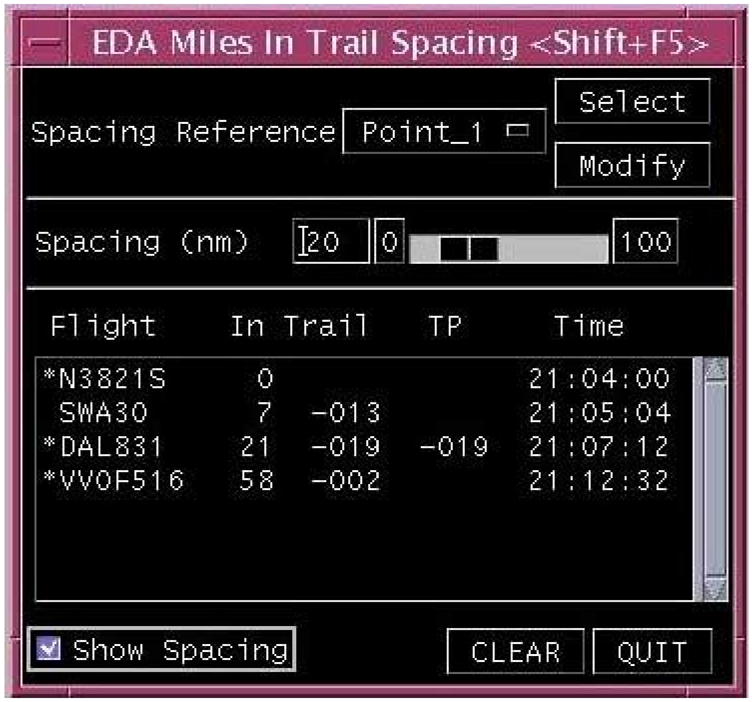


Figure 2-1: The Configured Miles-In-Trail Spacing Panel

2.2.1 Definition of "Miles-In-Trail"

Distance from the Spacing Reference is defined unambiguously as the distance of an aircraft *from its own particular crossing point at the time of the lead aircraft's crossing*, not the "in-trail" separation between aircraft in the Stream. This distance is the distance along the aircraft's trajectory.

Since spacing references may be defined as two-dimensional objects (Lines and Arcs), it is possible – even likely – that aircraft will cross the Spacing Reference at different locations and not be aligned in a "stream" per se.

2.3 Software Design Issues

2.3.1 Design Constraints

Insofar as is practicable, Seagull has endeavored to identify and make use of the software routines and design rationale that already exist in the CTAS baseline.

2.3.2 Software Reuse

Seagull identified existing software routines and requirements to be reused in the design and implementation of the Build 1 release. Among those most useful to our design were the following:

- Functions used to create various interface widgets.
- Functions that facilitate picking and graphic definition of the Spacing Reference directly from the PGUI canvas.
- Functions related to spacing, conflict detection and resolution, and trial planning.

- The functions used by the Profile Selector (Center) to update PGUI information at regular intervals.

2.3.3 Documentation Constraints

The En Route Descent Advisor Build-1 Final Report was prepared using the Microsoft Office 97 suite of tools.

2.3.4 Software Constraints

In compliance with NASA requirements, all software was coded using the C programming language. User interfaces were coded in X/Motif.

2.3.5 Performance Constraints

So far as practical, the new EDA software does not compromise current CTAS performance.

2.3.6 Security Constraints

All software testing involving live radar data feeds was performed at NASA Ames.

2.4 The Software Development Process

Seagull satisfied all NASA/CTAS requirements in regards to the software development process. The two principle development engineers at work on Build-1 have completed CTAS University at the Ames Research Center and made use of the DDTS and ClearCase tools over the lifetime of the development and integration process.

2.4.1 ClearCase Configuration Management

Seagull used the ClearCase configuration management tool for code development and management.

2.4.2 Design Defect Tracking System

Seagull used DDTS for problem and enhancement reporting.

2.4.3 CTAS University

As was previously noted, Seagull development engineers have successfully completed CTAS University at the Ames Research Center in preparation for this development effort.

2.4.4 The Development Environment

Seagull developed the EDA software on a Sun Sparc Workstation under the Solaris 2.6 operating system.

2.4.5 Baseline Integration

As far as practical and with the help of CTAS Software personnel, Seagull performed extensive testing of the EDA tool to ensure defect-free integration with the existing CTAS baseline. Such testing included running unrelated CTAS modules (TGUI, for example) to detect and remove side effects resulting from the EDA integration, if any.

3 Identification of Spacing Reference

3.1 Problem Definition

Aircraft in the Spacing Stream are spaced to a reference whose characteristics are defined by the operator before processing begins. The Miles-In-Trail Spacing tool includes an interface for this purpose, the Modify Spacing Reference (MSR) panel shown in Figure 3-1. From this panel, the operator may define a Point, Line or Arc spacing reference and specify its constituent parameters in a few simple steps. All reference definitions will become named objects and will be included in the menus provided on both panels. These references are available for use without redefinition as long as the tool is running, but do not persist after the tool is shut down. In future enhancements, storage of defined references on disk will be supported. An interface will be provided that allows the operator to specify which of the stored references should be included for a particular run. These references might be pre-sorted by sector, type, center, or some other unique characteristic.

The following sections describe the methods for defining a Spacing Reference and including flights into a Spacing Stream.

3.2 GUI Process Specification

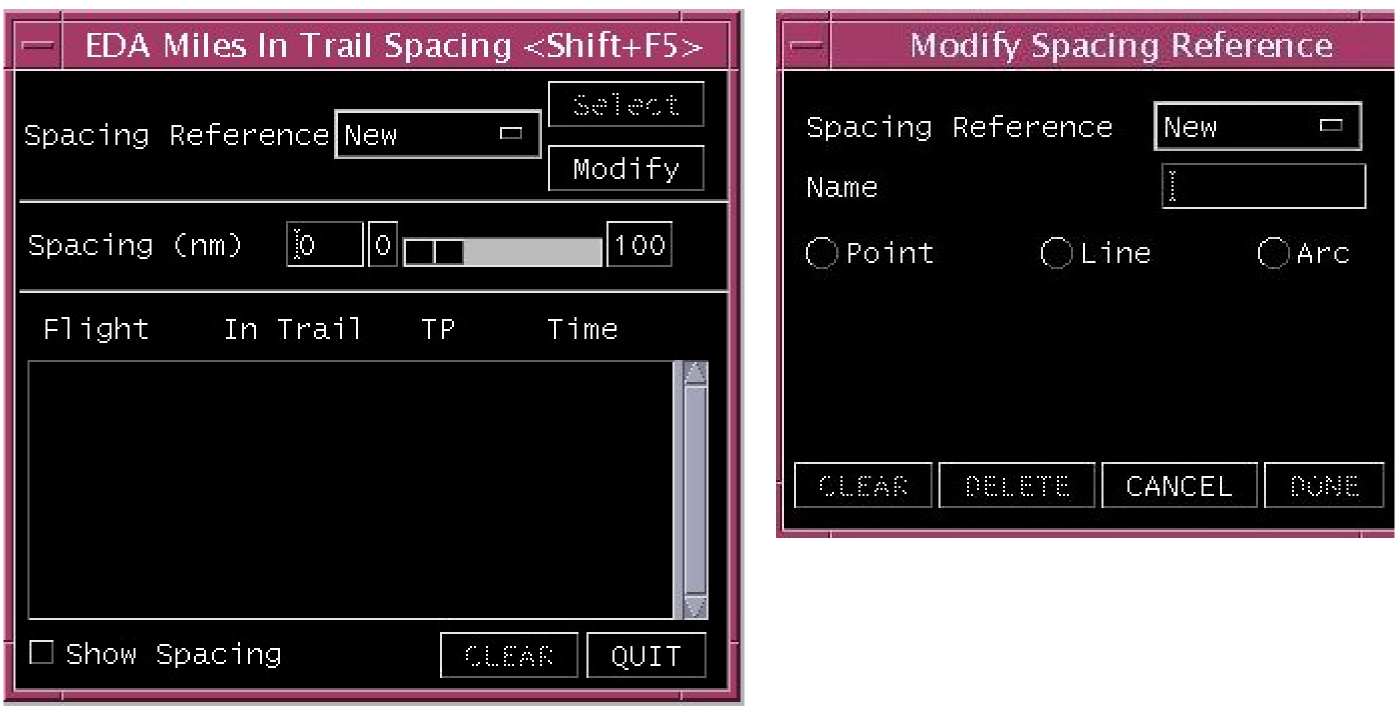


Figure 3-1 The Miles-In-Trail Spacing and Modify Spacing Reference Panels

3.2.1 Overview of the EDA Panels

When the PGUI is started in –upr mode, pressing <Shift+F5> summons the Miles-In-Trail (MIT) panel (shown on the left in Figure 3-1) to the screen in its initial, unconfigured state. The panel contains a pull down menu of defined Spacing References (initially empty), control buttons, a slider for specifying in-trail separation, and an area for displaying the Spacing Stream. Adjacent to the spacing reference pull-down list are two buttons labeled *Select* and *Modify*. Clicking the *Select* button sets the Spacing Reference that is displayed on the pull-down as the current reference and launches the spacing process, unless the current menu index is *New*. In the case of a *New* reference, the Modify Spacing Reference (MSR) panel, shown on the right in Figure 3-1, is displayed in its initial, unconfigured state, ready for the operator's input.

When an index other than *New* is selected, clicking the *Modify* button will cause the MSR panel to appear preconfigured with whatever parameters were used to define that Spacing Reference, and the operator may make modifications at will.

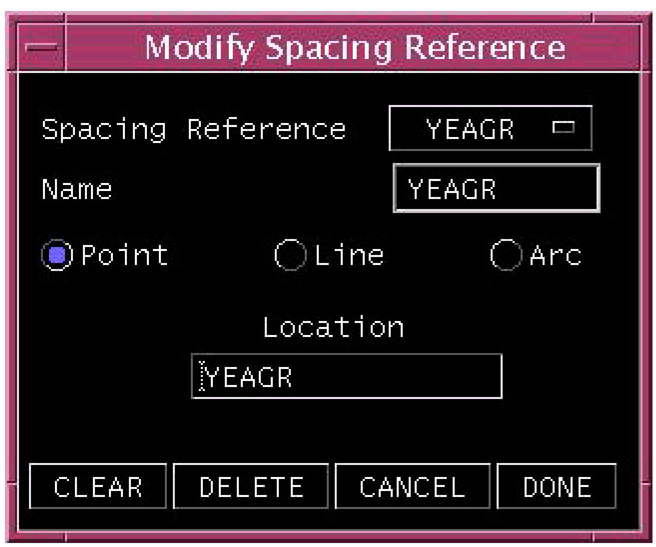
The MSR panel contains the following widgets:

Spacing Reference Menu: As described above, this menu contains all previously defined Spacing References. The menu is initially empty, except for the *New* option, which cannot be deleted.

Name Field: This field contains the name of the reference being created or modified. The operator can specify a value here, or the system will generate a unique value for a newly created reference. This is the name by which the spacing reference will be known.

Reference Type Radio Button: Labeled *Point, Line,* and *Arc*, the panel's radio buttons summon individual sub-panels configured for defining the desired reference type. Fields relevant to the choice will be displayed, and include (depending on reference type) *Location, Start, End,* and *Center Pt.* The fields are provided for either manual entry of named waypoints used in defining a reference, or to display the latitude/longitude location of points picked from the PGUI canvas.

The *Arc* sub-panel contains a slider for use in specifying the reference's desired radius. Examples of configured *Point*, *Line* and *Arc* sub-panels are shown in Figure 3-2.



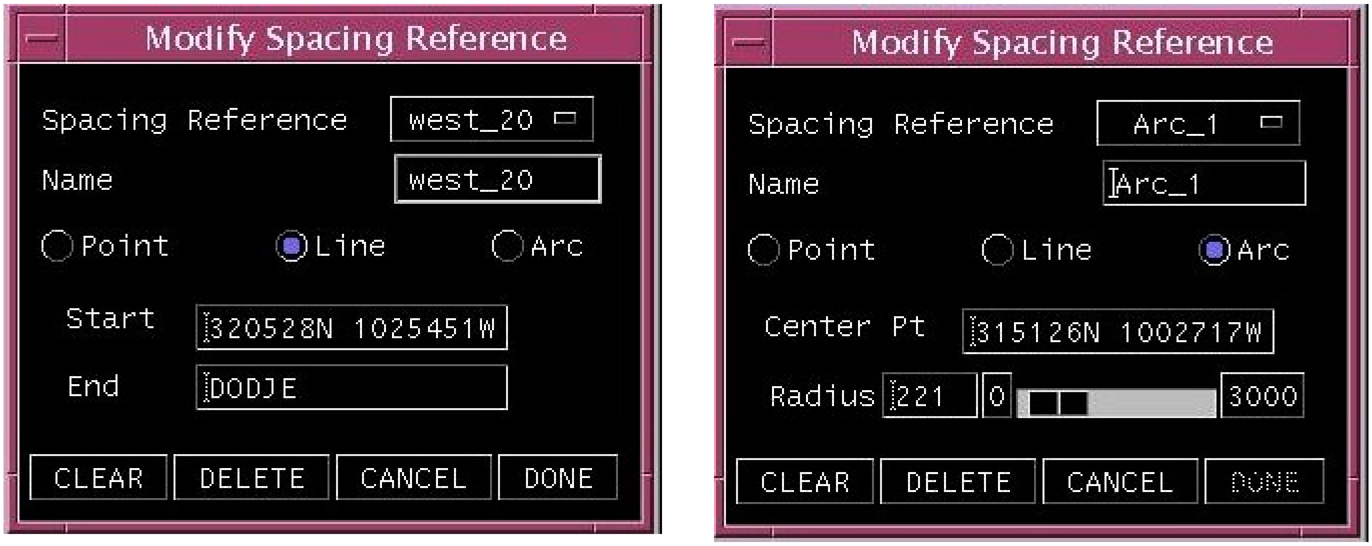


Figure 3-2 Configured Point, Line and Arc MSR Sub-Panels

Control Buttons: Arranged along the bottom of the panel are four control buttons: *CLEAR, DELETE, CANCEL,* and *DONE.* Refer to APPENDIX B, User's Guide: for more information on the functionality associated with these buttons.

3.2.2 Methods for Spacing Reference Definition

The operator may use one of two methods for defining the characteristics of a Point or Arc reference:

1. Entering a named waypoint into the appropriate MSR text field; or
2. Positioning the cursor on the PGUI and performing one of the following actions:
   a) Dwelling on a named waypoint and clicking, thereby selecting that waypoint; or
   b) Clicking anywhere on the canvas to "hook" an arbitrary latitude/longitude location.

For a Line reference, the operator sets an "anchor" or start point on the PGUI by mousing down, then drags to define the end point at some location not coincident with the start point. Alternatively, he or she may manually enter a named waypoint in either or both of the *Start* and *End* text fields. Any combination of named waypoint and latitude/longitude pair is permitted for a Line reference.

All definitions of references on the PGUI canvas is accomplished using the middle mouse button. Once the Modify Spacing Reference window is active, the middle mouse button is reserved exclusively for defining spacing references. Specifically, the ZOOM functionality normally associated with the middle mouse button is suspended until the operator clicks the *DONE* or *CANCEL* button on the MSR panel. Other mouse buttons are unaffected while the MSR is in operation.

3.2.3 Deleting a Spacing Reference

If a defined Spacing Reference is no longer required, it can be removed from the list of references by clicking on the MSR panel's *DELETE* key when that reference is displayed.

3.2.4 Future Spacing Reference Types

It is the opinion of the Seagull Design Team that in the future other spacing reference types could be added to the three defined above using the basic software building blocks used to support Point, Line, and Arc references. Two new spacing reference types seem obvious:

Polygonal Line Reference. This two-dimensional reference type would resemble a Line reference, but would consist of multiple sequential line segments. It would be useful, for example, to define sector or FIR boundaries as a spacing reference.

Composite References. This reference type could consist of multiple reference objects, perhaps multiple Point fixes. It would be useful in controlling a Spacing Stream converging on two or more Spacing References. There seems no logical reason why the components could not be any of the other reference types, e.g. Points, Lines, Arcs, or polygons.

4 The Spacing Stream

4.1 Spacing Stream Specification

Feedback about flights included in the Spacing Stream is the end product of the Miles-In-Trail Spacing Tool. Figure 4-1 shows information about a typical Stream displayed in the Spacing List area of the MIT panel.

Figure 4-1: The Spacing List on the MIT Panel

Data in the Spacing List will be updated approximately every 6 seconds to provide near real-time situational awareness for the operator. Data will be updated at a faster rate if Trial Planning is occurring. Aircraft selected to be part of the Spacing Stream will be differentiated from other aircraft on the PGUI canvas by the presence of a salmon colored box around their position markers as shown in Figure 4-2.

Figure 4-2 The Spacing Stream on the PGUI

4.1.1 Selecting a Spacing Reference for Processing

The *Spacing Reference* menu index displays the name of the Spacing Reference currently in use. The operator may initiate processing on any defined reference in the list by displaying the desired menu index and hitting the *Select* button. The system then recalculates the Spacing Stream values with respect to this new Spacing Reference.

The *Modify* button is used if the operator wishes to view or change the characteristics of the selected reference. A new reference is defined by positioning the Spacing Reference menu to the *New* index and clicking the *Modify* button. This summons the Modify Spacing Reference panel, and definition of the reference may begin as described in the previous sections and in APPENDIX B, User's Guide.

4.1.2 Selecting The In-Trail Spacing Value

As shown in Figure 4-3, manipulating the *Spacing (nm)* sliding scale sets the desired spacing value. Alternatively, the user may simply enter the target value in the text box to the left of the sliding scale.

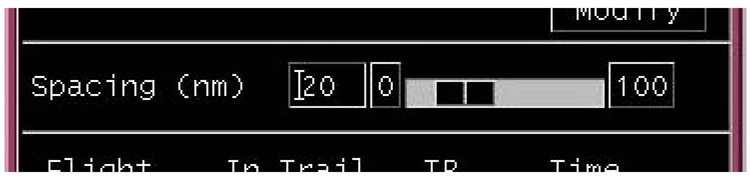


Figure 4-3: Selecting the In Trail Separation Value

4.1.3 The Spacing List Data Display

The scrollable Spacing List consists of five columns: aircraft ID, distance from the Spacing Reference, delta distance from desired spacing (positive or negative), trial planned delta distance from desired and estimated reference crossing time.

Figure 4-4: The Spacing List Data Display

The process for selecting aircraft into the Spacing Stream is covered in Section 6.

The aircraft are sorted in ascending order by distance from the Spacing Reference. Aircraft that have already passed the Spacing Reference but which have not yet been deleted from the Spacing List by the operator are at the top of the list. At the bottom of the list are aircraft that are moving away from, or on a trajectory that will never cross the Spacing Reference. Normally, it would be meaningless to select an aircraft that is moving away from the reference, but, if selected, the flight will appear at the bottom of the list.

Each of the fields in the Spacing List is described in the sections below.

4.1.3.1 Aircraft ID

The aircraft ID is the call sign or tail number of the aircraft. If the crossing point as calculated in Section 5 is greater than a specified configurable distance D from the Spacing Reference, the Aircraft ID will be preceded by an asterisk (*). Currently, this value is set as 20 nm in the pfs_defs.h header file in the profile_selector_cntr directory. The application of this rule depends on the type of reference defined by the operator. If the reference is a Point, the point of closest approach must be greater than D. If it's a Line, the intercept point of the trajectory with the extended line must be farther than D from the closest end point. If the reference point is an Arc, the aircraft will never be "out-of-bounds" once it crosses the Arc's boundary.

4.1.3.2 Distance from Spacing Reference

This field shows the calculated distance that each aircraft will be from the Spacing Reference when the lead aircraft crosses. If the aircraft is diverging from or will never cross the Spacing Reference, this field is set to dashes (-----). The algorithm necessary to calculate this value is described in Section 5. The distance being displayed is the distance along the aircraft trajectory to the crossing point.

4.1.3.3 Delta Distance from Desired

This field shows how far off the aircraft is from its desired spacing. For example, if the desired spacing is 20 nautical miles and an aircraft is third in line to cross the Spacing Reference its desired spacing is 40 nautical miles. If its actual spacing is 38 nm, a "–002" will be displayed in this column. If its spacing is 42 nm, a "+002" will be displayed.

4.1.3.4 Trial Planned Delta Distance from Desired

While an aircraft in the spacing list is being trial planned, its delta distance will be displayed in the fourth column. Once the trial plan is accepted or canceled, this column will return to the normal empty state.

4.1.3.5 Estimated Spacing Reference Crossing Time

This field shows the estimated time (in GMT) that each aircraft will cross the Spacing Reference. If the aircraft has already crossed the fix, this field contains the time of actual crossing. If the aircraft is diverging from the fix, this field is set to dashes (-----). The algorithm necessary to calculate this value is described in Section 5.

The definition of spacing reference crossing should be defined for each type of spacing reference:

Point: The aircraft is considered to cross the reference at its closest approach to the Point; i.e., the reference is in a position perpendicular to the trajectory of the aircraft.

Line: The aircraft is considered to have crossed the reference when it crosses the (extended) Line.

Arc: The aircraft is considered to have crossed the reference when it crosses the Arc boundary (represented as a circle at the given radius.)

4.1.4 Clearing the Spacing Stream

Clicking on the *CLEAR* button at the bottom of the Spacing List will remove all aircraft from the Spacing Stream and all spacing markers from the PGUI canvas.

4.1.5 Window Management

The Miles-In-Trail Spacing panel can be moved to any position on the PGUI. As with all other CTAS windows, both EDA panels are resizable, but will not re-scale when stretched.

Also as with all other CTAS windows, neither panel may be dismissed by clicking on its window menu button (upper left frame corner), or by selecting "Close" from that menu. Select <Shift+F5> to hide the panel(s), or *QUIT* to dismiss the panel(s) and end processing.

4.1.6 Toggling the Spacing Window

Pressing <Shift+F5> toggles the Miles-In-Trail Spacing panel (and the MSR panel, if it is displayed) off and on.

4.1.7 Quitting the Spacing Window

Selecting the *QUIT* button on the MIT panel will display a quit confirmation dialog box. Select *Confirm* to exit, *Cancel* to return to processing.

Figure 4-3 Exiting from EDA Processing

5 Algorithmic Process Specification

The Miles-In-Trail algorithm reports the distance that each aircraft is from the spacing reference when the first aircraft in the spacing stream crosses the spacing reference. Figure 5-1 illustrates a simple spacing scenario. The purpose of this section is to describe the algorithmic process to determine aircraft spacing. The section is broken down into a high level description of the algorithm followed by sections developing specifics of the algorithm. The final section briefly describes the relationship between these algorithms and the CTAS code.

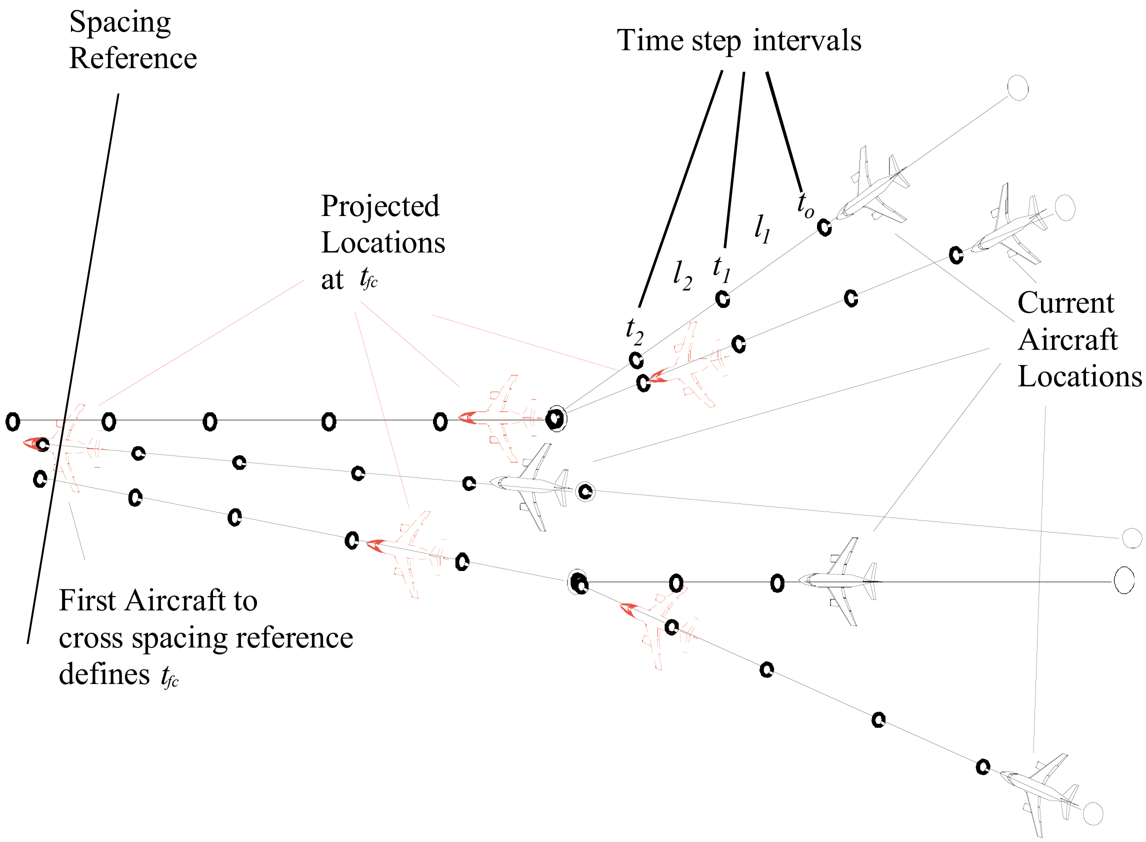


Figure 5-1 Illustration of traffic flow scenario analyzed by Miles-In-Trail Algorithm

5.1.1 Development of High Level Logic for the Miles-In-Trail Algorithm

The Miles-In-Trail algorithm starts at the current real time, noted by t_o, and performs an analysis of all m aircraft in the spacing stream and their respective trajectories. It predicts each aircraft's state at discrete time steps, Δt, into the future so that each aircraft's location is known at future instances all the way up to the time when the specific aircraft crosses the spacing reference. During this process, the algorithm determines the exact instant that each aircraft crosses the spacing reference. After determining the times at which each aircraft crosses the spacing reference, the algorithm finds which aircraft crosses the spacing reference first. The time associated with the first aircraft is noted as t_{fc}, the time of first crossing. A flowchart of the algorithm is shown in Figure 5-2. The nomenclature associated with the analysis is as follows:

- x_c, y_c, t_c: The ordered pair of coordinates and the time associated with the crossing of the spacing reference for a given aircraft.
- t_{fc}: The time of first crossing. This time corresponds to t_c for the aircraft that crosses the spacing reference first. All aircraft then use it as a reference time.
- x_{fc}, y_{fc}: The ordered pair of coordinates for a given aircraft that correspond to the time of first crossing, t_{fc}.
- d_c: The distance from a given aircraft's location at t_{fc} to the spacing reference
- n: The number of time steps between the current time and the time that a given aircraft crosses the spacing reference. (The value is different for each aircraft)
- Δt: The size of the time step e.g. 10 sec

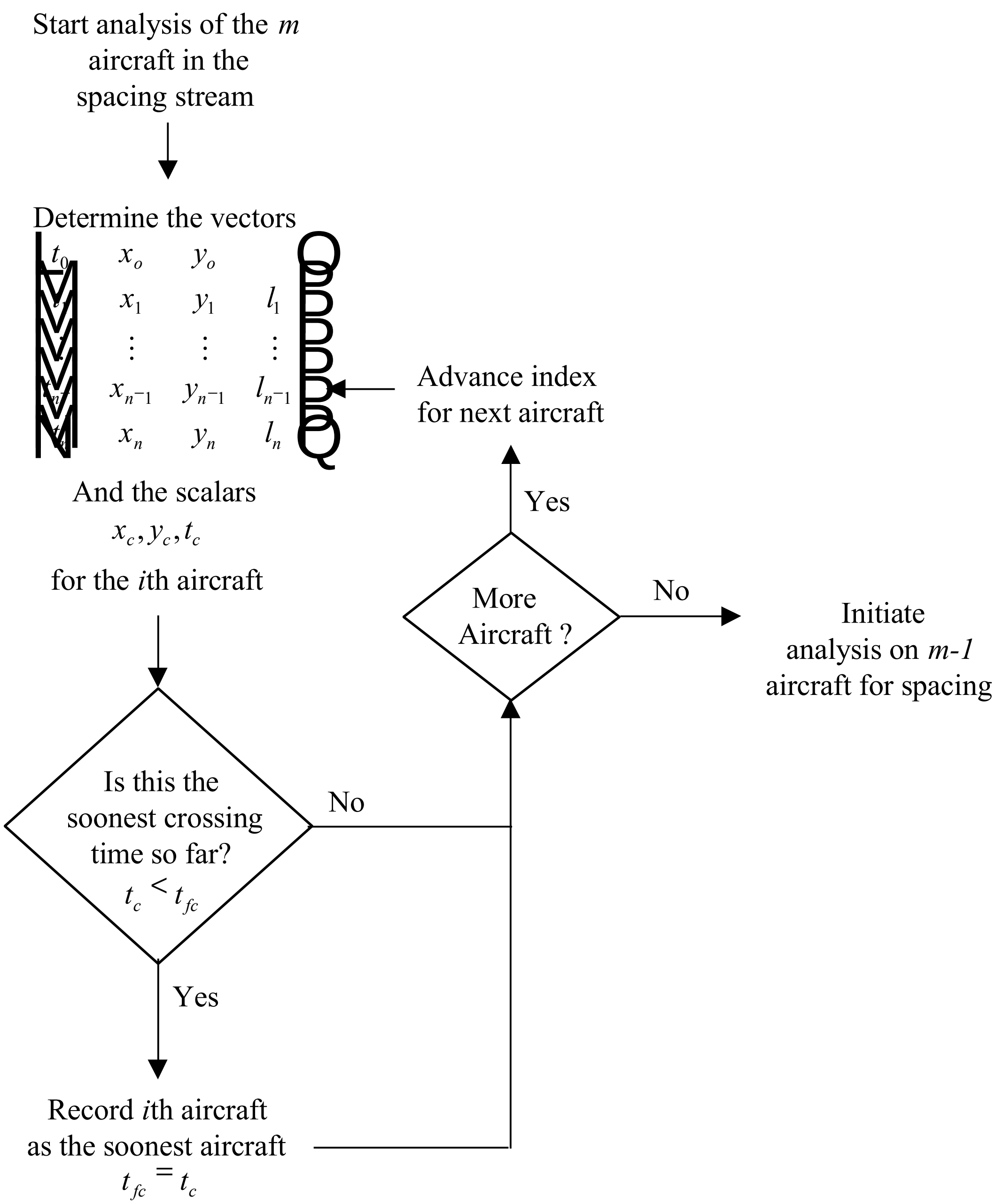


Figure 5-2 A flow diagram of the initial algorithm functionality which calculates vectors of aircraft state information, the crossing time and associated state information for each aircraft, and the time of first crossing, t_{fc}

The next step in the algorithm is to continue the analysis for $m-1$ aircraft that are not the first to cross the spacing reference as shown in Figure 5-3.

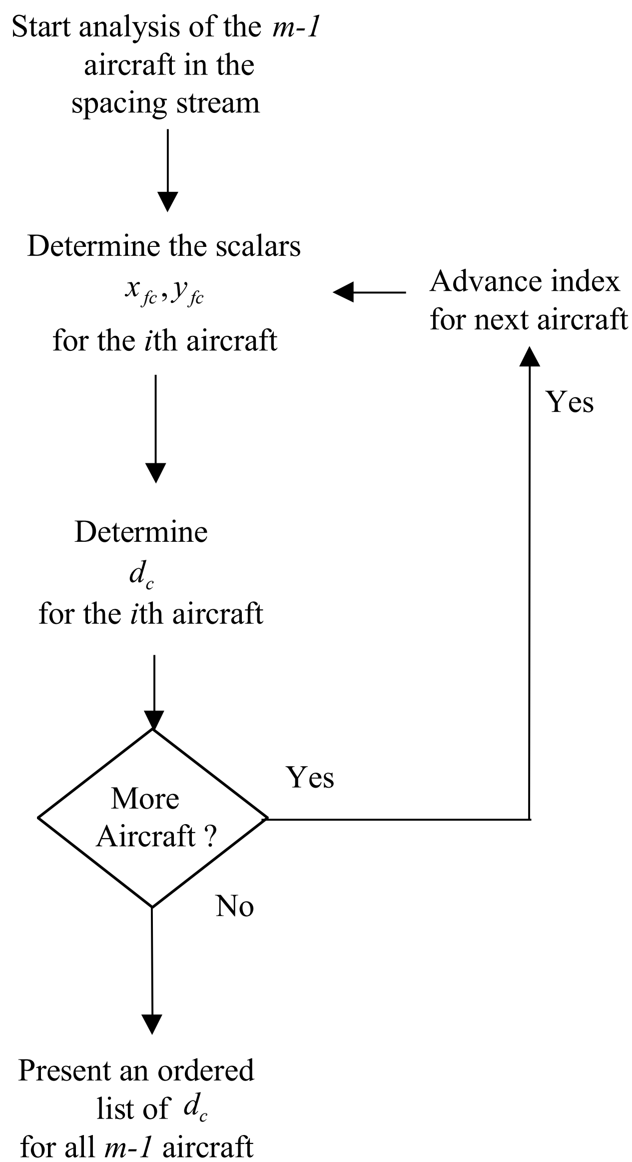


Figure 5-3 Algorithm for determining the spacing list

For each of these aircraft, the point along their trajectory that corresponds to the time t_{fc} must be determined. Then, the distance each aircraft is from the spacing reference can be calculated.

5.1.2 Algorithmic Development of Equations for Determining Distance and Time to a Spacing Reference

To support the functionality illustrated in Figure 5-2, the trajectory analysis routine steps along an aircraft's trajectory in a small time increments and estimates the aircraft's position at each interval as illustrated in Figure 5-4.

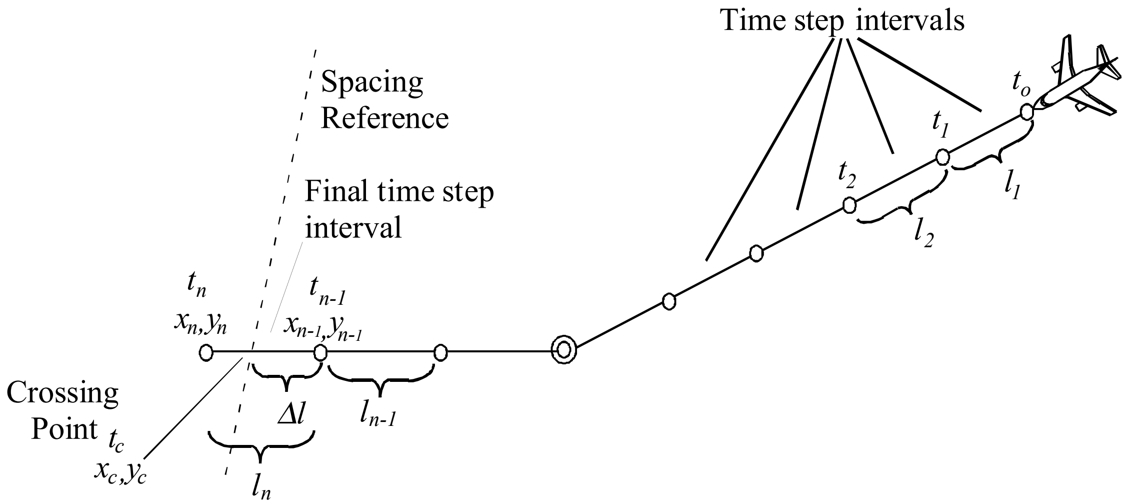


Figure 5-4 Illustration of algorithmic progression in small time steps along a trajectory

The data from the trajectory algorithm can be represented in vector format as shown in Equation (5.1) where the vectors exist for every aircraft in the spacing stream. The aircraft's location at each time step is stored along with the incremental distance covered during the interval.

$$\begin{bmatrix} t_o & x_o & y_o & \\ t_1 & x_1 & y_1 & l_1 \\ \vdots & \vdots & \vdots & \vdots \\ t_{n-1} & x_{n-1} & y_{n-1} & l_{n-1} \\ t_n & x_n & y_n & l_n \end{bmatrix} \tag{5.1}$$

The term l indicates the distance that the aircraft traveled in the time interval. The analysis terminates when the algorithm determines that the spacing reference has been reached. For purposes of the discussion, the *nth* time interval corresponds to the last time interval analyzed that is the crossing interval for the given aircraft. The distance that the aircraft travels along the route from t_o to the spacing reference is the summation of all of the incremental lengths. However, because the aircraft reaches the spacing reference before completely traversing the final segment, the final length is not l_n but rather Δl, the distance between t_{n-1} and the spacing reference. This is illustrated in Equation (5.2).

$$l_{total} = l_1 + l_2 + \cdots + l_{n-2} + l_{n-1} + \Delta l \tag{5.2}$$

In addition to the vector of trajectory information generated, explicit information about the crossing point for each aircraft must be determined. The required three parameters are x_c, y_c, and t_c as illustrated in Figure 5-4. The terms x_c and y_c are determined using Δl and a vector describing the final segment, $\vec{r}_r$ as defined in Equation (5.9). The term $\vec{r}_r$ can be broken down into its components as shown in Equation (5.3) and then x_c and y_c are solved directly using Equations (5.4) and (5.5).

$$\vec{r}_r = r_{r_x}\hat{x}_e + r_{r_y}\hat{y}_e \tag{5.3}$$

$$x_c = \frac{\Delta l}{|\vec{r}_r|} r_{r_x} \tag{5.4}$$

$$y_c = \frac{\Delta l}{|\vec{r}_r|} r_{r_y} \tag{5.5}$$

Similarly, the final time of crossing the spacing reference is shown in Equation (5.6) where V_{GS} is the ground speed of the aircraft along the segment.

$$t_c = t_{n-1} + \frac{\Delta l}{V_{GS}} \tag{5.6}$$

To determine the three important terms, the interval where the aircraft crosses the spacing reference must be found. Once the interval that crosses the spacing reference has been found, the algorithm concludes its search and focuses on this final time step. To support Equations (5.4) through (5.6), Δl must be determined. The distance from the crossing point to the spacing reference, d_{sr}, is also found. This section develops algorithms to determine these parameters for all three types of spacing references. All algorithms assume a flat earth reference frame.

5.1.2.1 Single Point Spacing References

The single point spacing reference represents the case where one reference fix is used to space aircraft. This fix can be on the trajectories of the aircraft to be spaced, but is does not have to be.

5.1.2.1.1 Determining which Segment Crosses the Spacing Reference

The analysis consists of a vector analysis of the segment location with respect to the spacing reference. We define several terms as follows:

1. $\vec{r}_n$: A vector from the head of the nth route segment to the spacing reference.
2. $\vec{r}_{n-1}$: A vector from the tail of the nth (head of the n-1) route segment to the spacing reference.
3. $\vec{r}_r$: A vector that describes the nth route segment.
4. x_n, y_n: An ordered pair describing the location of the aircraft at t_n in flat earth. coordinates
5. x_{n-1}, y_{n-1}: An ordered pair describing the location of the aircraft at t_{n-1} in flat earth coordinates.
6. x_{sr}, y_{sr}: An ordered pair describing the location of the reference fix in flat earth coordinates.

The vectors are defined explicitly in terms of the ordered pairs as shown in Equations (4.7) through (5.9). The terms illustrated in Figure 5-5.

$$\vec{r}_n = (x_{sp} - x_n)\hat{x}_e + (y_{sp} - y_n)\hat{y}_e \tag{5.7}$$

$$\vec{r}_{n-1} = (x_{sp} - x_{n-1})\hat{x}_e + (y_{sp} - y_{n-1})\hat{y}_e \tag{5.8}$$

$$\vec{r}_r = \left(x_n - x_{n-1}\right)\hat{i} + \left(y_n - y_{n-1}\right)\hat{j} \tag{5.9}$$

The vector analysis used to determine whether the *nth* segment crosses spacing reference point, compares several dot products of vectors

1. Condition which indicates spacing reference lies in front of current segment.

$$\begin{aligned} \vec{r}_n \cdot \vec{r}_r &> 0 \\ \vec{r}_{n-1} \cdot \vec{r}_r &> 0 \end{aligned} \tag{5.10}$$

2. Condition which indicates spacing reference crossing.

$$\begin{aligned} \vec{r}_n \cdot \vec{r}_r &< 0 \\ \vec{r}_{n-1} \cdot \vec{r}_r &> 0 \end{aligned} \tag{5.11}$$

3. Condition which indicates segment is beyond spacing reference.

$$\begin{aligned} \vec{r}_n \cdot \vec{r}_r &< 0 \\ \vec{r}_{n-1} \cdot \vec{r}_r &< 0 \end{aligned} \tag{5.12}$$

The when the spacing reference lies in front of the current segment, both vectors $\vec{r}_n$ and $\vec{r}_{n-1}$ are in the same general direction of $\vec{r}_r$. Therefore the dot products illustrated in Equation (5.10) will both be positive. This situation is illustrated in Figure 5-5. However, when the *nth* segment is abeam the spacing reference, the $\vec{r}_n$ vector will tend to point in the opposite direction of $\vec{r}_r$ which yields a negative dot product. This situation is illustrated in Figure 5-6. The condition where dot products are negative indicates that the aircraft has passed the spacing reference.

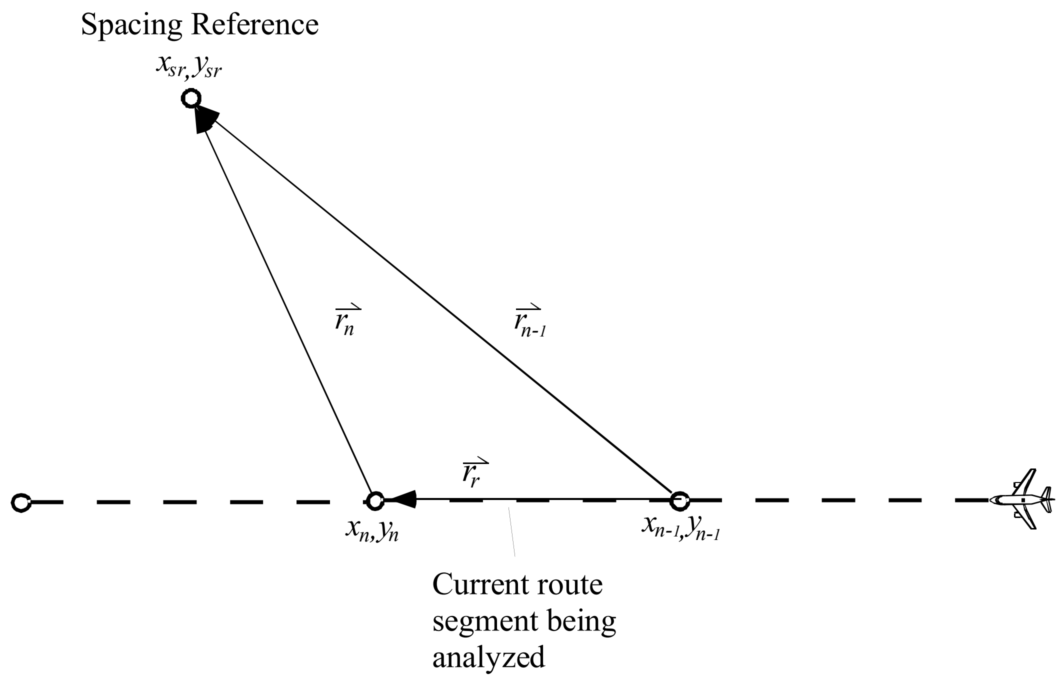


Figure 5-5 Illustration of a single point reference where the segment being analyzed does not cross the reference

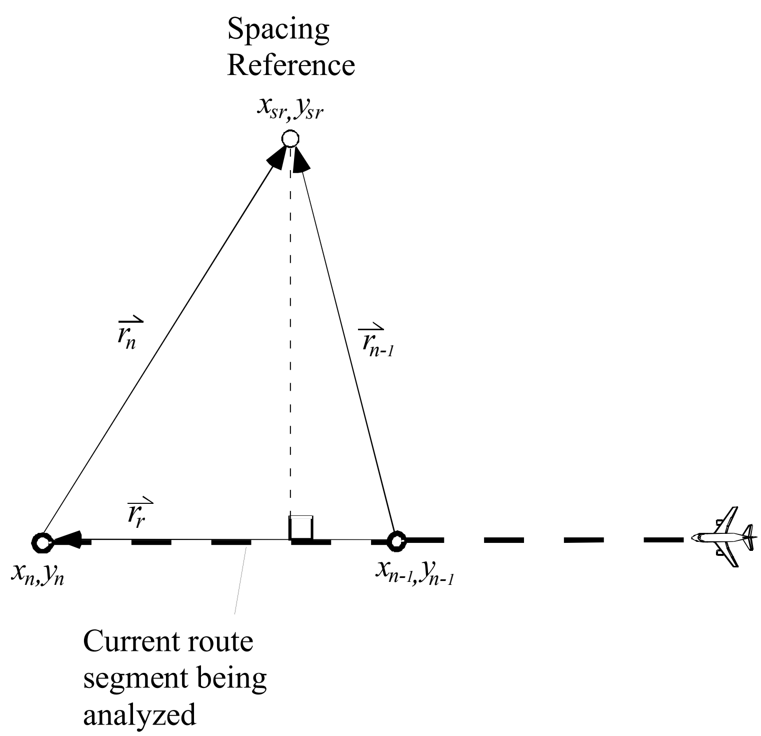


Figure 5-6 Illustration of a single point reference when the spacing reference lies abeam to the nth route segment

5.1.2.1.2 Determining the Final Leg Distance Δl and the Distance from the Spacing Reference

Figure 5-7 illustrates the geometry associated with the calculation of Δl. From observation, it is seen that if $\vec{r}_{n-1}$ is dotted with a unit vector in the direction of $\vec{r}_r$, the distance Δl is the result.

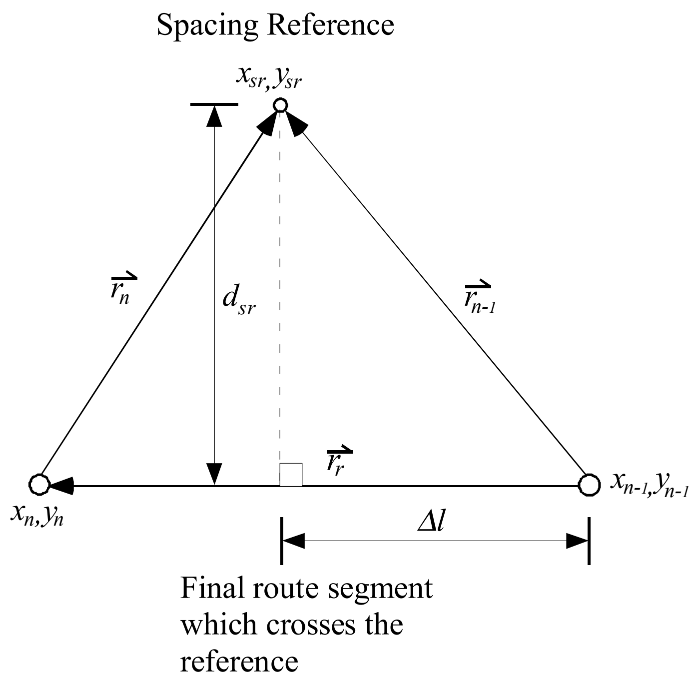


Figure 5-7 Illustration of geometry associated with the calculation of Δl and d_{sr}

This relationship is captured in Equation (5.13).

$$\Delta l = \vec{r}_{n-1} \cdot \frac{\vec{r}_r}{|\vec{r}_r|} \tag{5.13}$$

To determine the aircraft's distance from the spacing reference, we use Equation (5.14).

$$d_{sr} = \sqrt{|\vec{r}_{n-1}|^2 - \Delta l^2} \tag{5.14}$$

5.1.2.2 Line Spacing References

A line spacing reference is described by two fixes that are joined by a line. The line between the two segments is then extrapolated to continue in either direction infinitely. To properly characterize the line spacing reference we must choose an arbitrary point on the line, usually one of the initial fixes, a vector of arbitrary length that points along the line, and a unit vector that is normal to the line. The arbitrary point on the line becomes the spacing reference point and has a similar use in the line spacing as it did in the single point spacing. Furthermore, it is preferred that the unit vector normal to the line points in the general direction of the traffic flow, but it is not required. The nomenclature is defined as follows:

- x_{sr}, y_{sr}: An ordered pair of coordinates describing the first spacing fix used to describe the line. This pair will also become the arbitrary point on the line.
- x_{sr_2}, y_{sr_2}: The second ordered pair of coordinates describing the second spacing fix used to describe the line.
- $\vec{r}_{sr}$: A vector originating from the pair x_{sr}, y_{sr}, pointing along the length of the line.
- $\hat{n}_{sr}$: A unit vector normal to the line.

The terms $\vec{r}_{sr}$ and $\hat{n}_{sr}$ are expressed explicitly in Equations (5.15) and (5.16).

$$\vec{r}_{sr} = \left(x_{sr_2} - x_{sr}\right)\hat{x}_e + \left(y_{sr_2} - y_{sr}\right)\hat{y}_e \tag{5.15}$$

$$\hat{n}_{sr} = \frac{-\left(y_{sr_2} - y_{sr}\right)}{\sqrt{\left(x_{sr_2} - x_{sr}\right)^2 + \left(y_{sr_2} - y_{sr}\right)^2}}\hat{x}_e + \frac{\left(x_{sr_2} - x_{sr}\right)}{\sqrt{\left(x_{sr_2} - x_{sr}\right)^2 + \left(y_{sr_2} - y_{sr}\right)^2}}\hat{y}_e \tag{5.16}$$

The three vectors $\vec{r}_n$, $\vec{r}_{n-1}$, and $\vec{r}_r$ are required again, and are very similar to the point spacing reference definition.

- $\vec{r}_n$: A vector from the head of the nth route segment to the spacing reference point
- $\vec{r}_{n-1}$: A vector from the tail of the nth (head of the n-1) route segment to the spacing reference point
- $\vec{r}_r$: A vector which describes the nth route segment

When the direction of traffic is known, the sign on $\hat{n}_{sr}$ can be changed to insure that it points in the direction of the traffic flow. This proper sign could be determined by pointing the vector in the general direction of the arrival airport.

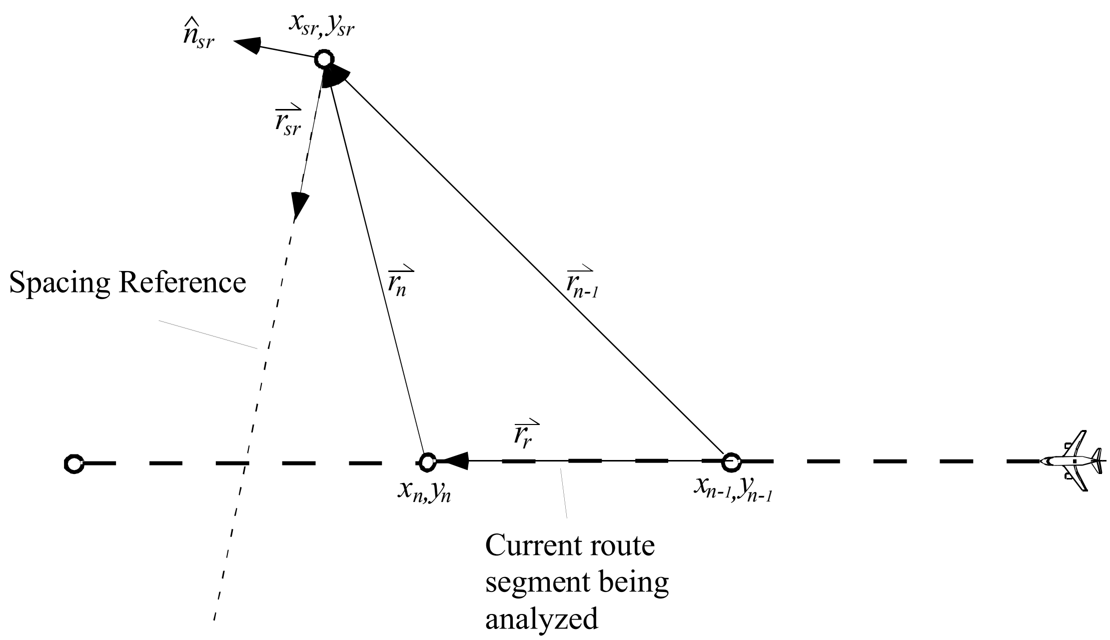


Figure 5-8 Illustration of line spacing analysis when current route segment does not cross the spacing reference

5.1.2.2.1 Determining which Segment Crosses the Spacing Reference

Determining which segment crosses the line spacing reference is similar to the point spacing algorithm except that $\hat{n}_{sr}$ is substituted for $\vec{r}_r$ in the dot product relations. Consider the case where the segment to be analyzed is does not cross the spacing reference as shown in Figure 5-8. In this situation, the observation shows that the two vectors, $\vec{r}_n$ and $\vec{r}_{n-1}$, will produce positive (or at least the same sign) results when dotted with $\hat{n}_{sr}$ because both vectors tend to point in the same direction as $\hat{n}_{sr}$. When the segment straddles the line spacing reference, the signs on the two dot products will be different because the $\vec{r}_n$ vector will tend to point in the opposite direction of $\hat{n}_{sr}$. This situation is illustrated in Figure 5-9.

The complete set of conditions is itemized below (Conditions assume $\hat{n}_{sr}$ points towards traffic flow):

1. Condition that indicates spacing reference lies in front of current segment.

$$\begin{aligned} &\vec{r}_n \,?\hat{n}_{sr} > 0 \\ &\vec{r}_{n-1} \,?\hat{n}_{sr} > 0 \end{aligned} \tag{5.17}$$

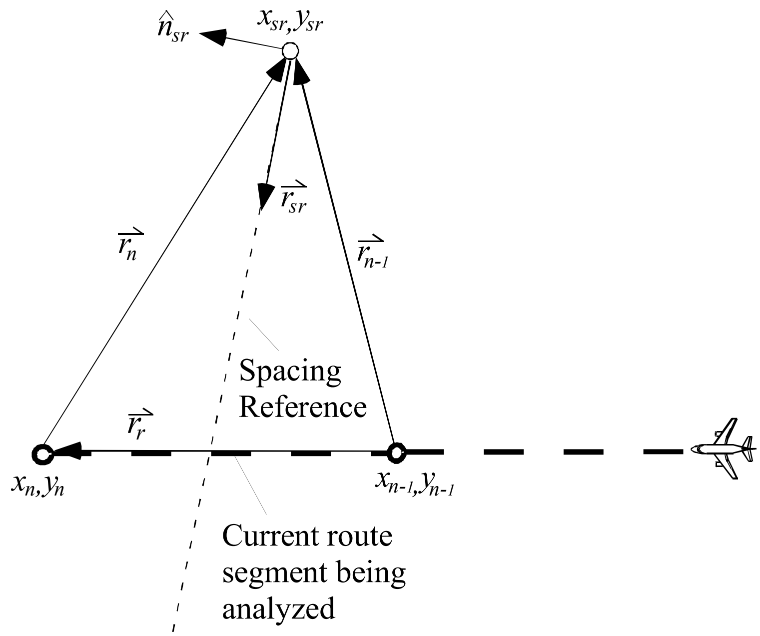


Figure 5-9 Illustration of line spacing analysis when current route segment crosses the spacing reference

2. Condition which indicates spacing reference crossing.

$$\begin{aligned} \vec{r}_n \cdot \hat{n}_{sr} &< 0 \\ \vec{r}_{n-1} \cdot \hat{n}_{sr} &> 0 \end{aligned} \tag{5.18}$$

3. Condition that indicates segment is beyond spacing reference.

$$\begin{aligned} \vec{r}_n \cdot \hat{n}_{sr} &< 0 \\ \vec{r}_{n-1} \cdot \hat{n}_{sr} &< 0 \end{aligned} \tag{5.19}$$

5.1.2.2.2 Determining the Final Leg Distance Δl and the Distance from the Spacing Reference

Determining the final leg distance Δl is more difficult with the line spacing reference because the aircraft trajectory is not guaranteed to cut the segment at right angles. This creates inconvenient geometry. Consider the geometry shown in Figure 5-10.

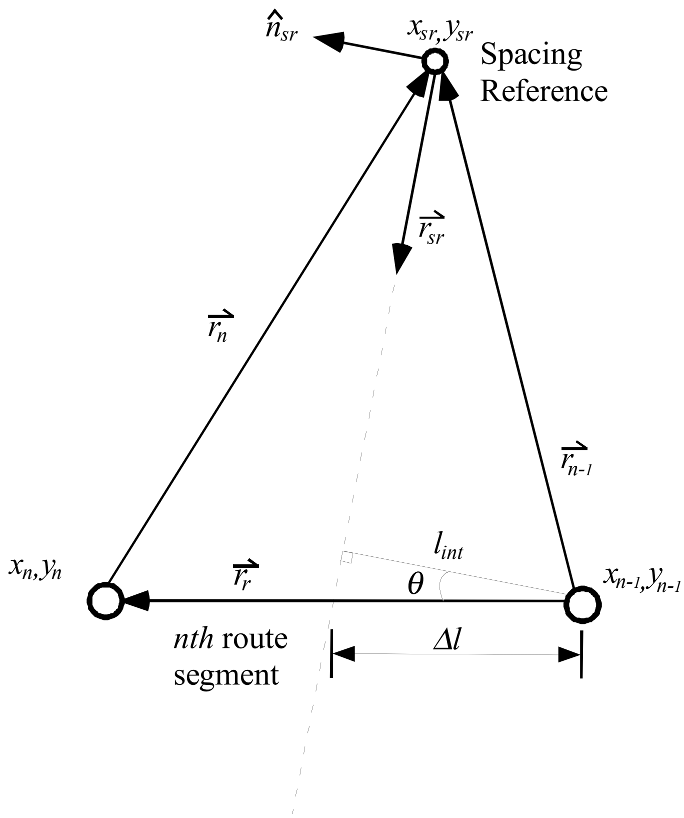


Figure 5-10 Illustration of geometry used to determine Δl

To determine Δl, we must first recognize that the dot product of $\hat{n}_{sr}$ and $\vec{r}_{n-1}$ is equal to l_{int}, an intermediate term defined as shown in Figure 5-10.

$$l_{\text{int}} = \vec{r}_{n-1} ?\hat{n}_{sr} \tag{5.20}$$

Furthermore, we can see the trigonometric relationship between l_{int} and Δl as expressed in Equation (5.21).

$$l_{\text{int}} = \Delta l \cos\theta \tag{5.21}$$

However, the angle θ is unknown. We know that θ is the angle between $\hat{n}_{sr}$ and $\vec{r}_r$ so an expression for θ can be determined using the dot product definition as shown in Equations (5.22) and (5.23).

$$\vec{r}_r ?\hat{n}_{sr} = |\vec{r}_r||\hat{n}_{sr}|\cos\theta \tag{5.22}$$

$$\cos\theta = \frac{\vec{r}_r ?\hat{n}_{sr}}{|\vec{r}_r||\hat{n}_{sr}|} \tag{5.23}$$

Combining Equations (5.20), (5.21), and (5.23) we have Equation (5.24) and finally Equation (5.25).

$$\left(\frac{\Delta l}{|\vec{r}_r|}\vec{r}_r\right)\cdot\frac{\vec{r}_{sr}}{|\vec{r}_{sr}|} \tag{5.24}$$

$$\Delta l = |\vec{r}_r|\,|\hat{n}_{sr}|\frac{\vec{r}_{n-1}\cdot\hat{n}_{sr}}{\vec{r}_r\cdot\hat{n}_{sr}} \tag{5.25}$$

Determining the distance from the spacing reference, d_{sr}, is more complicated with a line. The geometry for determining d_{sr} is shown in Figure 5-11. The case where the aircraft is in the positive direction of $\vec{r}_{sr}$, is different than the case when the aircraft is on the negative direction of $\vec{r}_{sr}$.

Part of the distance is comprised of the dot product of $\vec{r}_{n-1}$ with a unit vector in the direction of $\vec{r}_{sr}$. The final tip of the distance is determined by the dot product of a vector of length Δl in the direction of $\vec{r}_r$ with unit vector in the direction of $\vec{r}_{sr}$. The result is shown in Equation (5.26) where we prime d_{sr} to denote that the sign of d_{sr} has not yet been considered.

$$d_{sr}' = -\vec{r}_{n-1}\cdot\frac{\vec{r}_{sr}}{|\vec{r}_{sr}|} + \left(\frac{\Delta l}{|\vec{r}_r|}\vec{r}_r\right)\cdot\frac{\vec{r}_{sr}}{|\vec{r}_{sr}|} \tag{5.26}$$

If we combine Equations (5.25) and (5.26), we can eliminate Δl as shown in Equation (5.27).

$$d_{sr}' = \left(\left(\frac{\vec{r}_{n-1}\cdot\hat{n}_{sr}}{\vec{r}_r\cdot\hat{n}_{sr}}\right)\vec{r}_r - \vec{r}_{n-1}\right)\cdot\frac{\vec{r}_{sr}}{|\vec{r}_{sr}|} \tag{5.27}$$

If d_{sr}' is negative, the appropriate value for d_{sr} is simply the absolute value of d_{sr}'. If d_{sr}' is positive, the length of the line segment must be subtracted. Of course, if d_{sr}' is smaller than the line segment length, the aircraft is considered to be at the spacing reference.

For negative d_{sr}'

$$d_{sr} = |d_{sr}'| \tag{5.28}$$

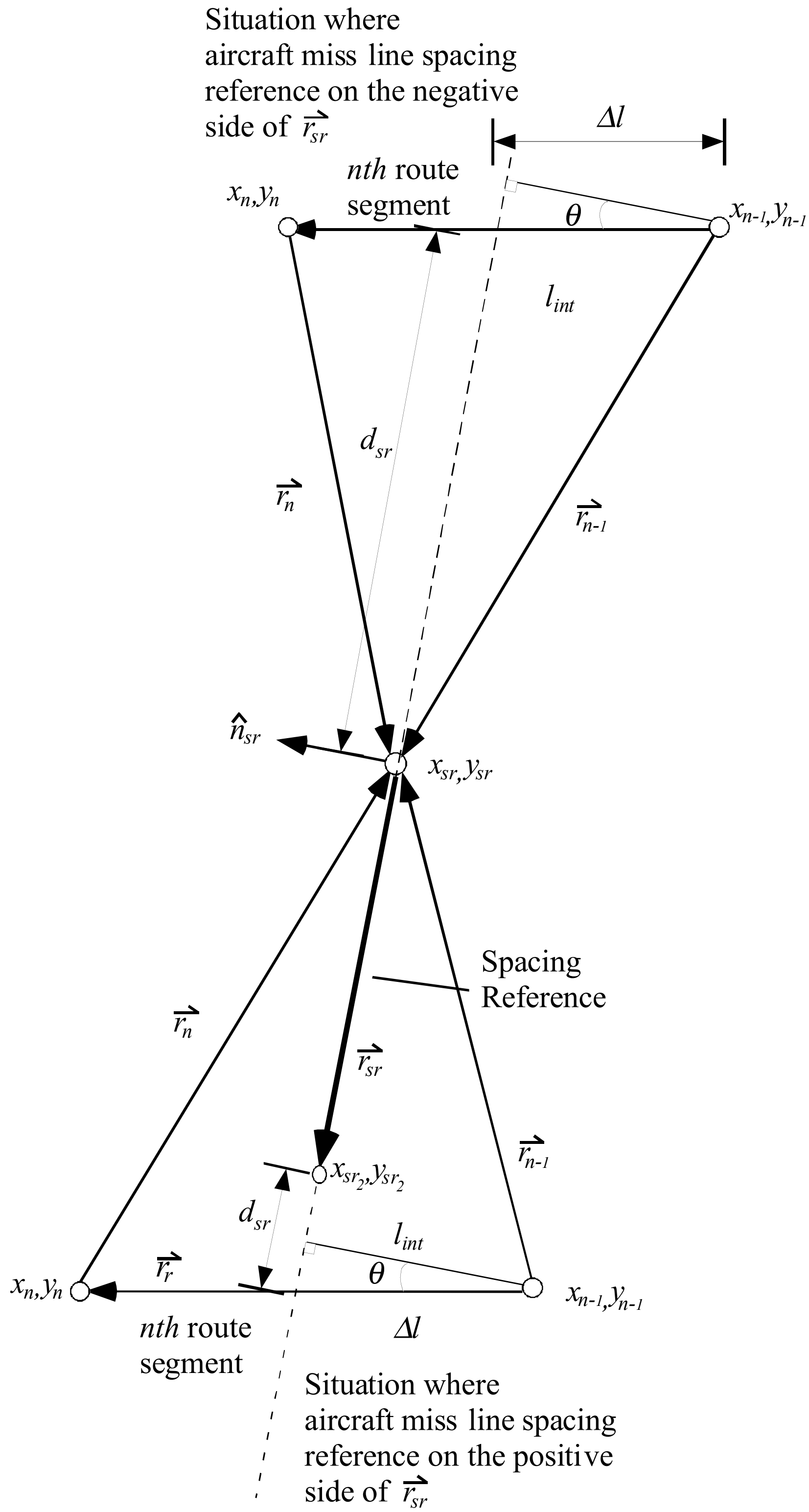


Figure 5-11 Geometry associated with the distance from the spacing reference for a line spacing reference

For positive d_{sr}'

$$d_{sr} = d_{sr}' - \left|\vec{r}_{sr}\right| \tag{5.29}$$

5.1.2.3 Arc Spacing References

Arc spacing references used in this algorithm are all circular which greatly simplifies the analysis. Determining the crossing segment is easy because it requires simply a comparison of distance with no vector manipulation. Determining Δl is more complex because the intersection between the segment and the circle must be determined exactly. To manipulate the arc spacing reference, we need to define some new terms. These are

1. x_{sr}, y_{sr}: An ordered pair describing the location of the center of the arc in flat earth coordinates
2. r_a: The radius of the arc.

5.1.2.3.1 Determining the Crossing Segment for the Arc Spacing Reference

Consider the geometry in Figure 5-12. We can see that anytime the magnitude of $\vec{r}_n$ is smaller than the radius of the arc, r_a, the segment has crossed the Arc. We express it mathematically in Equation (5.30).

1. Condition for crossing Arc Spacing Reference

$$\left|\vec{r}_n\right| < r_a \tag{5.30}$$

We can express it with a scalar expression as shown in Equation (5.31)

$$\sqrt{\left(x_n - x_{sp}\right)^2 + \left(y_n - y_{sp}\right)^2} < r_a \tag{5.31}$$

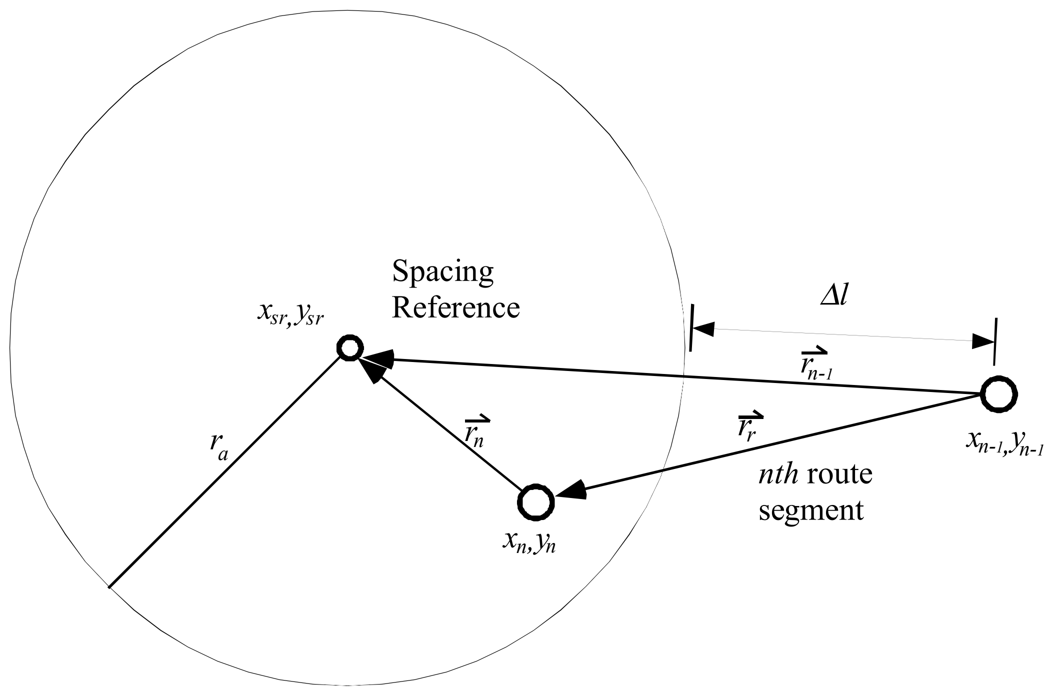


Figure 5-12 The geometry associated with the arc spacing reference

5.1.2.3.2 Determining the Final Leg Distance Δl

Determining the final leg distance is more difficult it requires determining the intersection point between the line segment and the circle. First we define the intersection point as the ordered pair (x_i, y_i). Any point along the circle must conform to the equation of a circle as shown in Equation (5.32).

$$(x_i - x_{sp})^2 + (y_i - y_{sp})^2 = r_a^2 \tag{5.32}$$

Furthermore, we must express the *nth* line segment as an equation of a line (Equation (5.33) where the slope and y-intercept, m and b, respectively, are defined by Equations (5.34) and (5.35).

$$y_i = mx_i + b \tag{5.33}$$

$$m = \frac{(y_n - y_{n-1})}{(x_n - x_{n-1})} \tag{5.34}$$

$$b = y_n - mx_n \tag{5.35}$$

Combining Equations (5.32) and (5.33) we have Equation (5.36). After simplification, Equations (5.40) and is our final result.

$$(x_i - x_{sp})^2 + (mx_i + b - y_{sp})^2 = r_a^2 \tag{5.36}$$

$$x_i^2 - 2x_i x_{sp} + x_{sp}^2 + m^2 x_i^2 - 2m(b - y_{sp})x_i + (b - y_{sp})^2 = r_a^2 \tag{5.37}$$

$$x_i^2 + \frac{2m[b - y_{sp}] - 2x_{sp}}{(1 + m^2)} x_i + \frac{[b - y_{sp}]^2 + x_{sp}^2 - r_a^2}{(1 + m^2)} = 0 \tag{5.38}$$

$$x_i^2 + \frac{2\{m[b - y_{sp}] - x_{sp}\}}{(1 + m^2)} x_i + \frac{[b - y_{sp}]^2 + x_{sp}^2 - r_a^2}{(1 + m^2)} = 0 \tag{5.39}$$

$$x_i = \frac{\{x_{sp} - m[b - y_{sp}]\}}{(1 + m^2)} \pm \frac{1}{2}\sqrt{\left[\frac{2\{m[b - y_{sp}] - x_{sp}\}}{(1 + m^2)}\right]^2 - 4\frac{[b - y_{sp}]^2 + x_{sp}^2 - r_a^2}{(1 + m^2)}} \tag{5.40}$$

Since Equation (5.40) is quadratic, some answer checking will have to be done to insure the proper x_i is returned. The term y_i can then be easily determined using Equation (5.33). Using the point (x_i, y_i), the distance Δl is straightforward to determine using Equation (5.41).

$$\Delta l = \sqrt{(x_i - x_{n-1})^2 + (y_i - y_{n-1})^2} \tag{5.41}$$

5.1.3 Determining the Final Spacing Distances

Once the lead aircraft of the spacing stream has been determined and all the aircraft trajectory information has been compiled in vector form, it is possible to support the functionality illustrated in Figure 5-3. To determine the spacing list, the algorithm must first calculate x_{fc} and y_{fc}, and subsequently d_c, the distance each aircraft is from the spacing reference at time t_{fc}. The terms x_{fc} and y_{fc}, are calculated by using linear interpolation between the compiled vectors of aircraft trajectory information as illustrated in Figure 5-13.

$$\begin{array}{l}
\left[t_0 \quad t_1 \quad \ldots \quad \overbrace{t_{i-1} \quad \cdots \downarrow}^{t_{fc}} \quad t_i \quad \cdots \quad t_{n-1} \quad t_n\right] \\
\left[x_0 \quad x_1 \quad \ldots \quad x_{i-1} \quad \cdots x_{fc} \quad x_i \quad \cdots \quad x_{n-1} \quad x_n\right] \\
\left[y_0 \quad y_1 \quad \ldots \quad y_{i-1} \quad \cdots y_{fc} \quad y_i \quad \cdots \quad y_{n-1} \quad y_n\right]
\end{array}$$

Figure 5-13 Graphical illustration of the linear interpolation required to determine x_{fc} and y_{fc}

The *ith* index corresponds to the location in the time vector just past the t_{fc} term. Therefore the interpolation to determine x_{fc} and y_{fc} handled with Equations (4.42) and (4.43).

$$x_{fc} = x_{i-1} + \frac{t_{fc} - t_{i-1}}{t_i - t_{i-1}} \left(x_i - x_{i-1} \right) \tag{5.42}$$

$$y_{fc} = y_{i-1} + \frac{t_{fc} - t_{i-1}}{t_i - t_{i-1}} \left(y_i - y_{i-1} \right) \tag{5.43}$$

To determine d_c, the distance to the spacing reference, there are two competing approaches. These approaches are illustrated by Figure 5-14. The first approach is a simple line between points. The second is a summation of the intervals along the route. Equation (4.44) illustrates the first concept.

$$d_c = \sqrt{\left(x_c - x_{fc} \right)^2 + \left(y_c - y_{fc} \right)^2} \tag{5.44}$$

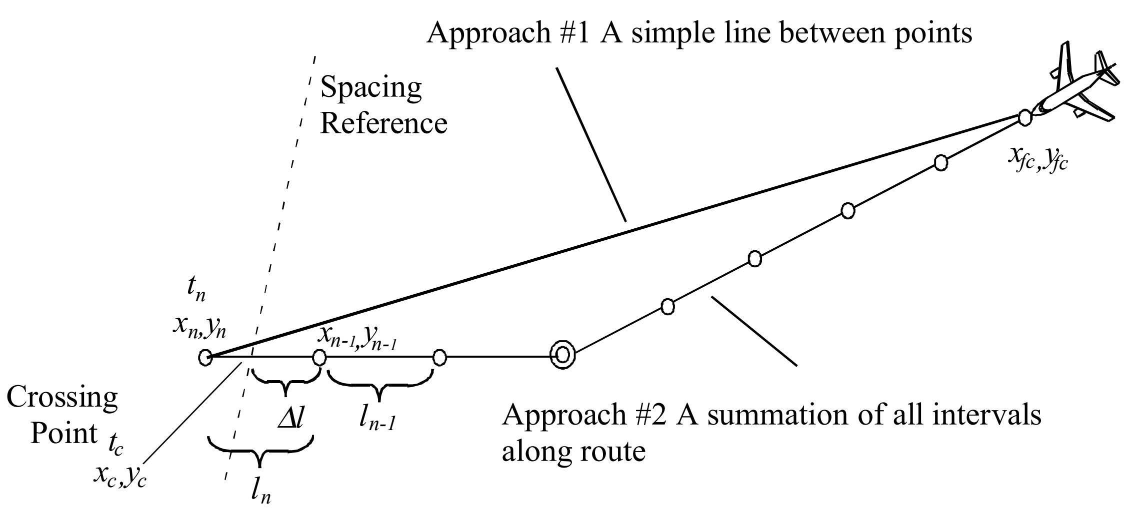


Figure 5-14 An illustration of the two different methods for calculating distance from the spacing reference

Equation (4.45) yields the second alternative that characterizes the actual route of flight.

$$d_c = \sqrt{\left(x_i - x_{fc} \right)^2 + \left(y_i - y_{fc} \right)^2} + l_i + \cdots + l_{n-2} + l_{n-1} + \Delta l \tag{5.45}$$

5.1.4 Relationship between Spacing Algorithm and CTAS code

The algorithms to determine the crossing point for each of the spacing references were implemented in the new spacing code. The CTAS code did contain a function to determine the time an aircraft crosses a waypoint; however, it does not include similar functions for crossing a line or arc. Therefore, to be consistent, the vector method was used for all types of references. The vector method also was helpful in determining if an aircraft will never cross a reference. And since using this method also produces the distance from the spacing reference, it was used to check to see if the aircraft was out-of-bounds when it crossed the reference.

The CTAS code does contain methods for finding distances given times or x and y points so these functions were used in place of adding up the lengths of the individual time steps.

6 Identification of Aircraft in the Spacing Stream

6.1 Problem Definition

The selection of aircraft into a spacing stream depends on the ATM purpose of the spacing list. For example, the list might be used to space traffic landing at a local airport with limited capacity, or it might be used to meter en route traffic destined for a distant center with weather problems.

6.2 GUI Process Specification

All aircraft displayed on the Planview GUI are eligible for inclusion into a spacing stream. In Build 1, selection and deletion of spacing stream aircraft will be done manually by the operator. However, Seagull made provision for the automatic selection of aircraft in some later phase and, to this end, a conceptual design has been formulated.

6.2.1 Manual Addition and Deletion of Aircraft in the Spacing Stream

Adding Aircraft to the Spacing Stream: For Build 1, the operator shall select an aircraft by placing the cursor over the aircraft type in the data tag displayed on the PGUI (Line 3, Field 'A') and clicking with the right mouse button. The selection of this new aircraft will cause the aircraft to appear in the MIT panel's Spacing List window. The addition of new aircraft will cause the spacing list to be recalculated so the new aircraft will appear in its proper sequence in the Spacing Stream.

Aircraft selected to be part of the spacing stream will be differentiated from other aircraft on the PGUI canvas by the presence of a salmon colored box around their position markers.

Deleting Aircraft from the Spacing Stream: For Build 1, the operator will delete aircraft manually from the Spacing Stream. The user will place the cursor over the aircraft type in the data tag of a previously selected aircraft (Line 3, Field 'A') and click with the right mouse button. This will normally occur after the aircraft passes the Spacing Reference but it can be done at any time. Aircraft may also be deleted from the Spacing Stream by deselecting the aircraft from the PGUI; that is, left clicking on the aircraft symbol.

Aircraft that have already passed the Spacing Reference are not automatically removed from the Spacing Stream. Rather, these flights will show dashes in the *In Trail* column of the MIT panel's Spacing List window and will persist at the top of the list until deleted by the operator.

6.2.2 Automatic Addition and Deletion from the Spacing Stream

Providing functionality to facilitate the automatic filtering of flights for inclusion in the Spacing Stream is not a requirement under Build 1; however, there is a requirement that Seagull consider ways in which this might be accomplished.

For subsequent builds of the EDA, we propose that an automatic aircraft selection mechanism be implemented for defining the Spacing Stream. An aircraft selection screen – or *Filter* - that would specify one or more criteria based on information in the aircraft's flight plan would control this. Flights entering the system would be tested against the Filter's criteria; if a match is found, the flight will be automatically added to the Spacing Stream. Such Filters could be named and stored for later reuse.

Selection Criteria may include information found in a typical flight plan, such as the following:

Departure Airport
Destination (and/or Alternate) Airport
En route Fix
En route Airway
Departure Center or Sector
En route Center or Sector
Destination Center or Sector
Altitude Range (hi and low)

Aircraft Type
Call Sign Filter (Airline Code)
Aircraft Type
Flight Plan Speed
Departure Time (Range)
Landing Time (Range)
Equipment Suffix
Center

Figure 6-1 Filter Definition for Automatic Selection

Figure 6-1 shows a possible design for the EDA Miles-In-Trail panel with the addition of the filter definition functionality; specifically, with the addition of a *Spacing Filter* menu and its associated *Select* and *Modify* buttons. Just as the *Spacing Reference* menu lists all pre-defined references, the *Spacing Filter* menu will list all pre-defined selection Filters that can be chosen by the operator. Processing will begin on whatever Filter is visible from the *Spacing Filter* menu when the operator hits the *Spacing Reference* menu's *Select* button; at any other time during processing, another Filter may be selected by choosing it from the *Spacing Filter* menu and then hitting *that* menu's *Select* button. The Filter *Modify* button will summon a panel like the one shown in Figure 6-2 in which the Filter's parameters may be specified or edited.

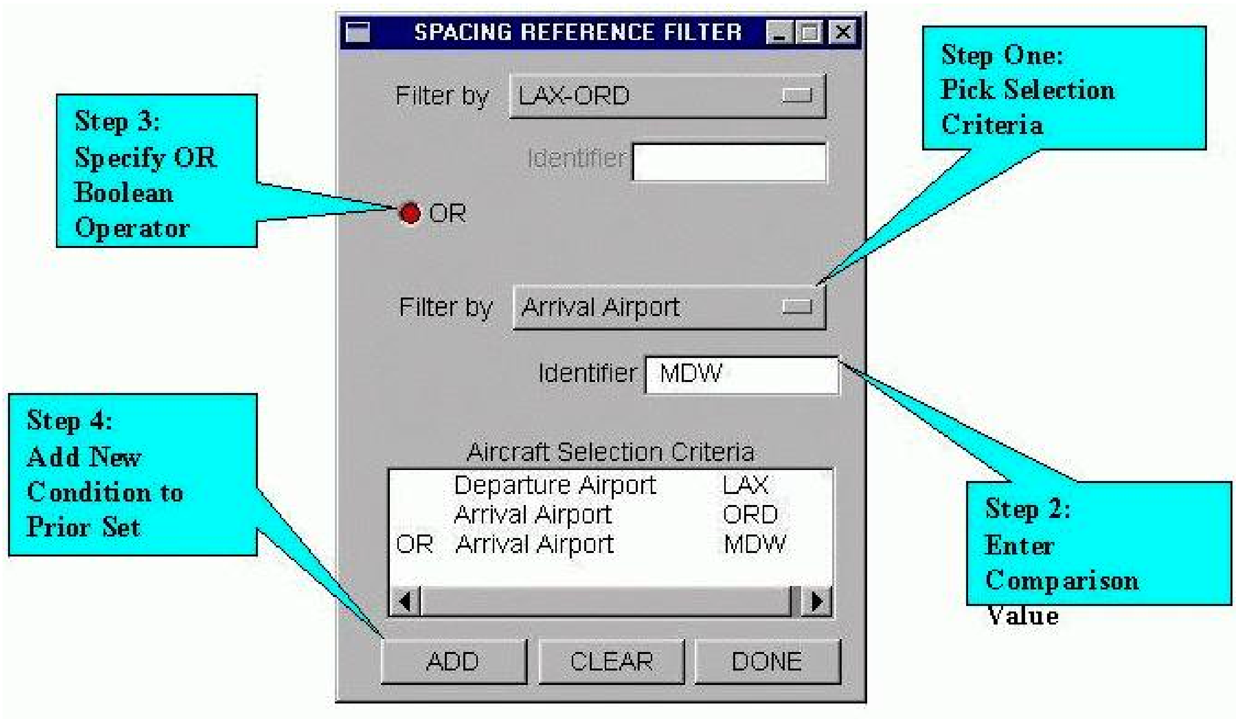


Figure 6-2 The Spacing Stream Filter Panel

7 CPTP Interaction with In-Trail Spacing

The In-Trail Spacing tool will not alter the Conflict Probe and Trial Planning (CPTP) functionality within CTAS. However, the Spacing Tool will interact with whatever trial planning and conflict probe capabilities currently exist in CTAS.

Once the user sees that a spacing distance has been violated by a flight, he or she may adjust the flight's trajectory by using the Trial Planning tool. While the operator is making trial adjustments (in flight path, altitude or speed – if possible in CTAS) to the trajectory, the flight's spacing distance will be updating. Also, as currently implemented, the new trajectory will be conflict probed. Once the user has decided on a new trajectory that satisfies the spacing and is conflict free, s/he selects the *ACCEPT* button in the Trial Planning window and the trial trajectory is now the active trajectory. This new active trajectory will then be checked for conflicts within the normal time cycle.

8 Issuing Clearances on Active Aircraft and In-Trail Spacing

When the user selects one of the spacing aircraft and issues changes in speed or altitude through a PGUI "hot" key, CTAS issues a clearance for the new aircraft. If the aircraft is on the Spacing List, the new trajectory will be used to calculate the new spacing distance. The user must "own" the aircraft to be able to issue the clearance.

9 Software Code and Process

For the development of the Spacing Tool, code and features that already exist in CTAS were used. The following is a summary of the features that were reused, as is or with some modifications, and those items that were added.

9.1 Set Up Window

The shift<F5> Key brings up the Miles-In-Trail Spacing Window.

New functionality -	new PGUI code for the new window.
Reuse -	code for raising window and creating window.

9.2 Select Spacing Reference

The user selects a predefined spacing reference, or defines a new one.

New functionality -	new data structure to hold spacing reference.
New functionality -	draw a line by click and drag.
Modified message -	the PGUI_DA_SPACING_TOGGLE message has been modified to contain the spacing reference.
New display -	to highlight Spacing Reference and color it salmon.
Reuse -	draw circle, window button, checkbox, string edit, pull down list, etc. functionality.

9.3 Set Spacing Distance

User sets distance to be used for In Trail Spacing.

New functionality -	set variable to be used in Miles-In-Trail Spacing Window. This variable will be used to identify aircraft that violate spacing value.
Reuse -	slider bar to set value.

9.4 Select Spacing Stream

User selects aircraft to be a part of the spacing stream.

New functionality -	new data to hold aircraft in spacing list but based on selected aircraft structure.
New Message -	create PGUI_SPACING_AIRCRAFT message.
New functionality -	draw a salmon colored box around aircraft symbol to highlight spacing aircraft.
Modify -	select item from data block to add or delete aircraft from spacing stream.
Reuse -	function to add/remove an aircraft to a list

9.5 Interprocess Communication between pgui and pfs_c

The structure that is used to exchange data between pgui and pfs_c was modified.

The new structure is defined as:

```
typedef struct Pgui_unique_info_st
{
  int                         fd;
  Spacing_toggle_type         spacing_tool_mode;
  int                         spacing_tool_waypoint;
  Spacing_reference_st        spacing_reference;
```

```
   Direct_to_toggle_type     direct_to_mode;
   Conflict_pred_toggle_type conflict_pred_mode;
   Selected_ac_list_st       selected_ac_list;
   Spacing_stream_st         spacing_ac_list;
   Conflict_pair_list_st     violation_list;
   Conflict_pair_list_st     previous_violation_list;
   Conflict_pair_list_st     unforced_violation_list;
   Force_conf_pred_list_st   force_conf_pred_list;
   Spacing_tool_output_st    spacing_tool_output[MAX_ADVISORIES];
}  Pgui_unique_info_st;
```

where, the new members are spacing_reference and spacing_ac_list and spacing_tool_output has been modified to include more data.

In pfs_read.c, read_msg( ) has been modified to handle the new spacing aircraft message and spacing reference message.

9.6 Calculation of Spacing Information in pfs_c

The set_spacing( ) function is called from Frames 2 and 8 (every 6 seconds) if a spacing reference has been set and if there is at least one aircraft in the spacing stream.

The set_spacing( ) function is called every second during Trial Planning if a spacing reference has been set and if the aircraft being trial planned is in the spacing stream.

The set_spacing( ) function in process_spacing.c has been modified to:

1. Handle different types of spacing references
2. Calculate spacing reference crossing time and distance for each aircraft relative to its own crossing
3. Set BOUNDS_LIMIT in pfs_defs.h to 20 nautical miles.
4. Not eliminate any aircraft from calculation but mark if out of bounds or unable to calculate.
5. Access trial planning trajectories when aircraft are being trial planned.

9.7 Display of Spacing Information on Miles-In-Trail Spacing Window

Once a PS_DA_SPACING_INFO message has been received by the pgui, the sorted list of spacing stream aircraft is displayed in the Miles-In-Trail Spacing window.

New functionality - A new data structure to hold spacing stream information, (e.g. acid, distance from lead aircraft, UTC time when crosses reference, flags) has been created.

New functionality - comparison of the distance from the desired distance.

9.8 Display Spacing Stream Projected Positions on Pgui

Once a PS_DA_SPACING_INFO message has been received by the pgui, a marker is drawn on the pgui to show the position on each spacing aircraft when the first aircraft crosses the spacing reference. Markers for aircraft that will never cross or have already crossed are not displayed.

New functionality - Color of marker has been changed to salmon.

9.9 Software Development Life Cycle Iterations

This sub-section outlines the order in which the software was developed.

9.10 PGUI

9.10.1 Prototyping

1. Prototype Miles-In-Trail Window
2. Prototype Spacing Reference Window
3. Learn how to compile and debug in CTAS environment.

9.10.2 Interaction With Miles-In-Trail Window

Combine pgui interaction with Miles-In-Trail and Spacing Reference Windows

1. Bring up blank Miles-In-Trail Window with shift<F> key
2. Select Spacing Reference
3. Draw Spacing Reference on PGUI
4. Develop interaction between PGUI and Modify Spacing Reference window
5. Select Spacing Stream
6. Identify Spacing Stream aircraft on PGUI

9.10.3 Fill Data Structures and Send Messages to CM

1. Create Spacing Reference structure.
2. Utilize messages to send Spacing Reference data to CM.
3. Utilize messages to send Spacing Stream data to CM

9.10.4 Receive Messages from CM and Fill Data Structures

1. Receive Spacing Output Message from CM.
2. Create and propagate data structure to hold spacing information if necessary.
3. Compare individual aircraft distance from previous aircraft to Miles-In-Trail value.

9.10.5 Display Spacing Stream

1. Display Spacing Stream data on Miles-In-Trail Window.
2. Display projected Spacing Stream data on PGUI.
3. Handle flights that cross out-of-bounds, never cross or have already crossed Spacing Reference.

9.10.6 Develop Strategy for Persistence of Spacing Reference

1. Plan what data need to be saved.
2. Plan how to save data.
3. Plan how to retrieve data.

9.11 PFS_C

9.11.1 Prototyping

1. Prototype crossing algorithms. .
2. Learn how to compile and debug in CTAS environment

9.11.2 Code

1. Code new set_spacing( ) function.
2. Code crossing algorithms.
3. Modify `Pgui_unique_info_st` to suit spacing needs.

9.11.3 Receive Message from CM and fill Data Structures

1. Use modified message to handle Spacing Reference.
2. Fill data structures needed for set_spacing( ) function

9.11.4 Call set_spacing( ) from Frames 2 and 8 of Update Loop

1. Determine if have enough information to call set_spacing( ).
   a. Must have a Spacing Reference defined
   b. Must have at least 1 aircraft in a spacing stream.
2. Add procedures to call set_spacing( ) from Frames 2 and 8 of update loop.

9.11.5 Calculate Spacing and Fill Output Data Structure

1. Test set_spacing( ) function with input data
2. Fill Spacing Tool Output structure with results

9.11.6 Send Message to CM

1. Fill spacing message with data from Spacing Tool Output structure.
2. Test sending spacing message to CM

9.11.7 Implement accessing set_spacing( ) from Other Updates.

1. Implement and test functionality of set_spacing( ) while Trial Planning is occurring.
2. Implement and test functionality of set_spacing( ) when a new clearance has been given for an active aircraft.

9.12 Ongoing Activities

9.12.1 nasa_branch Interaction

1. Create new branch from nasa_branch
2. Keep new branch current with nasa_branch
3. Merge back into nasa_branch after unit testing and code review by CTAS SW personnel.

9.12.2 Integration Testing

1. Test pgui and pfs_c together after individual testing.
2. Test with other CTAS tools

10 Possible Future Enhancements

The En route Descent Advisor is ripe for enhancement that will expand the functionality delivered with Build 1. The following are some possible new features.

10.1 Spacing References

10.1.1 Spacing Reference Display Toggle

Add ability to toggle between displaying the Spacing Reference as a named fix, as latitude / longitude, and X / Y value. Currently if the Point Reference, either of the endpoints of the Line Reference, or the center of the Arc Reference lies on a named waypoint, that name will be shown. Otherwise, the latitude and longitude of the point will be displayed.

10.1.2 Add Other Types of Spacing References

Add ability to define sector boundary and groups of named fixes as Spacing References.

10.1.3 Indicate if a Spacing Reference is off the Screen

It is possible to specify a Spacing Reference which lies somewhere outside the boundaries of the PGUI as it is currently configured. In this instance, a marker indicating the direction to the reference and its type will be displayed on the PGUI. For example, if the PGUI is centered on DFW and the operator defines an Arc Spacing Reference centered on ORD, the EDA software would position a marker in the upper right corner of the PGUI which details this information.

10.2 Selection and Display of Spacing References and Spacing Stream

10.2.1 Multiple Spacing References (SRs) and Spacing Streams (SSs)

Implement the ability to have multiple SRs and SSs. Investigate how to deal with 2 different SSs going through the same SRs.

This task would restructure the EDA, build 1 spacing tool to permit it to run multiple instances of the spacing tool on the same PGUI. Each spacing tool would have its own spacing reference and spacing stream. This task would involve both PGUI work and PFS_C for handling the different lists.

10.2.2 EDA Spacing Tool Broken up into Two Parts

Investigate breaking up the tool into "Master" and "Sector Controller" tools. The "Master" would be used by the TMU or the person running an experiment to set the fixed number SR-SSs. The sector controller would then be able to choose from the list of SR-SS and configure his/her display.

This is a major new concept that allows sharing of a spacing tool between different PGUIs. This task involves PGUI mostly. This task also includes the ability of the Sector controller to be able to configure his display. The handling of multiple references and streams is covered above.

9.2.3 Automatic Selection of the Spacing Stream (Aircraft Selection Filter)

Providing functionality to facilitate the automatic filtering of flights for inclusion in the Spacing Stream is not a requirement under Build 1; however, there is a requirement that Seagull consider ways in which this might be accomplished. Section 6.2.2 provides a brief summary of how selection filters might be implemented and what sort of interface could be provided for their definition and modification.

10.3 Spacing List Display and Use

10.3.1 Presentation of Spacing List

Investigate best way to present the spacing. Look at actual distances, delta distances, distance from requested, use of color vs. text, etc. Create various types of displays and be able to easily toggle between them for experiments.

10.3.2 Display Line of Sight or Path Distances

Allow the user to select which distance to display.

This task involves both PGUI and PFS_C. Both types of distances would have to be calculated and saved. Then the user could access the one desired.

10.3.3 Delete a Flight Directly from the Spacing List

Expand the Spacing List capabilities to be able to select a flight on the list and remove it with *DELETE* button or with the Delete key. This is currently not a function of a list in the CTAS library.

This would involve creation of a new CTAS list object with much more functionality than the current one.

10.4 Automation

10.4.1 Automatic Drop Criteria

Seagull and Ames should arrive at some "automatic drop criteria" for flights that have crossed the Spacing Reference. The TMU or person running the experiment could select these criteria. They may include, among others, such cases as "n-minutes after handoff," "immediately after fix crossing," and "n-miles after fix crossing".

10.4.2 Automatic Selection Criteria

Expand design of the filter presented in the Draft Design Document to implement automatic selection of aircraft into a Spacing Stream. Add ability to select a Center as one of the criteria.

This is a PGUI functionality enhancement.

10.5 EDA Performance Analysis

10.5.1 Analyze Performance of Spacing Tool

Determine what processing load the Spacing Tool imposes by running benchmark performance tests. Identify those functions, computations and loops that are heavy computation users.

10.5.2 Performance Enhancements

Optimize the heavy users of system resources.

10.5.3 Restructure Spacing Tool as Stand-Alone Process

With the prospect that the EDA Build 1 Spacing Tool could become widely used by CTAS (i.e, multiple spacing tools running at each PGUI), restructure the spacing tool computation functionality to run as a separate process communicating with the PGUI processes through the CM. This could allow the EDA to run, if needed, on a separate processor to meet performance requirements.

10.6 Other Items

10.6.1 Point of Closest Approach Spacing Probe

Investigate a Spacing Probe that will be able to identify the Point of Closest Approach between 2 aircraft. Design the best way to present this functionality to the user.

10.6.2 Management of Residual Error

Add the enhancement to allow the lead aircraft not to show a spacing of 0.0. Currently the error in the 2nd aircraft goes away when it becomes the 1st aircraft.

Design of this enhancement may be related to the automatic drop criteria of "immediately after fix crossing" and must be investigated with respect to that case. Need to fully understand exactly what the customer would like to see.

10.6.3 Problem of Trial Planning an Aircraft Far Away from Spacing Reference

Reduce the amount of clearances that a controller may have to give if trial planned an aircraft that is far down stream of the reference. Research this problem and solutions after input from customer.

10.6.4 Time Based Spacing List

The Build 1 Spacing Tool reports the “position” of aircraft in a Spacing Stream as miles behind a lead aircraft. This is appropriate if all the aircraft are traveling at around the same speed. With aircraft of greatly different performance, e.g. turbo prop vs. turbo jet, it would be useful to compare the relative position of the aircraft in time over the fix order rather than distance from the fix order.

This is a PGUI enhancement to allow user to toggle between versions of Spacing List.

10.6.5 Error Handling

Add functionality to check for user and computational errors and handle them cleanly and efficiently.

10.6.6 Data Logging

Add functionality to log data during a CTAS run.

10.6.7 Documentation

Document all new features.

APPENDIX A, Requirements Cross Reference Table

Req. Number	Requirement Statement	Final Report-Ref
2.3.1	Design Constraints	
2.3.1.1	Seagull shall identify existing software routines and algorithms that can be "reused" in the Build-1 effort.	9
2.3.1.2	Documents shall be written in Microsoft Office 95, or later.	
2.3.2	Software Constraints	
2.3.2.1	Seagull shall satisfy all NASA/CTAS requirements for developing and tracking code within the current CTAS baseline environment.	2.4
2.3.2.1.1	Seagull shall use the ClearCase configuration management tool for code development and management.	2.4.1
2.3.2.1.2	Seagull shall use DDTS for problem and enhancement reporting.	2.4.2
2.3.2.1.3	Seagull employees who are developing code shall be trained through the CTAS University system.	2.4.3
2.3.2.2	The software shall be written in the C or C++ programming language.	2.3.4
2.3.2.3	The software shall be developed on a Sun Sparc workstation under the Solaris 2.6 operating system	2.4.4
2.3.2.4	Seagull shall develop the Build-1 tool within the CTAS software baseline.	2.4.5
2.4	Non-Functional Requirements	
2.4.1	Deliverables	

2.4.1.1	Build-1 design document and briefing. Seagull shall deliver a written document, in hard copy and on electronic media (Microsoft Office 95, or later) detailing the results of the Build-1 design (sub-task 1). The document shall serve both as a useful reference to NASA/EDA researchers and as a road-map for the follow-on sub-task 2 effort. In addition, Seagull shall provide a briefing to NASA/EDA researchers summarizing the Build-1 design. This briefing shall serve as a formal kickoff to the sub-task 2, software development effort.	1.1
2.4.1.2	Build-1 final report. Seagull shall deliver a final report, in hard copy and on electronic media (Microsoft Office 95, or later), summarizing the capabilities and features implemented for EDA Build-1 under sub-task In particular, the document shall highlight any deviations from the document Build-1 design. The final report shall also include a *User's Guide* to the Build-1 prototype tool.	2.3.3
2.4.1.3	Build-1 software implementation. Seagull shall implement EDA Build-1 into the CTAS baseline.	2.4.5
2.4.2	Performance Requirements, The Build 1 software shall not compromise current CTAS capabilities and operate in a manner such that researchers can make meaningful assessments of the tool's functionality.	2.3.5
2.4.3	Security Requirements. While working with live data feeds, all development shall take place on site at NASA Ames.	2.3.6
2.5	Sub Task 1 Requirements – Build 1 Software Design Requirements . Create software design requirements and specifications for an En Route Descent Advisor (EDA) research tool that incorporates the identification of the spacing stream, the spacing fix and trial planning for spacing and conflict resolution. The design will also specify graphical and textual feedback.	2.2
2.5.1	Seagull system engineers shall develop a detailed understanding of the Build-1 en route spacing tool concept through existing NASA research papers and briefings, CTAS prototype demos, and interaction with NASA researchers responsible for EDA development.	1.1

2.5.2	The contractor shall determine functional design specifications for the EDA Build-1 tool sufficient to initiate the software development effort in sub-task 2.	2.2 Entire EDA Build-1 Design document
2.5.3	The design shall include both algorithmic and GUI design specifications.	3 4 5 6
2.5.4	In developing design requirements and specifications, the contractor shall assess the following features and characteristics: Identification of the spacing stream Identification of the spacing reference Trial-planning for spacing and conflict resolution	 6 3 7
2.5.4.1	Identification of the spacing stream	
2.5.4.1.1	The Build-1 tool shall allow for the identification/selection of aircraft for which a TFM-generated spacing requirement shall apply.	6
2.5.4.1.2	Algorithmic and GUI process specifications should be developed that consider both automated and manual techniques for aircraft/stream identification	6.2.1 6.2.2
2.5.4.1.3	Manual selection techniques shall be developed for Miles-In-Trail constraint input.	6.2.1
2.5.4.1.4	Specifications for manual techniques shall include the processes and procedures for allowing the controller to select and deselect individual aircraft for spacing.	6.2.1
2.5.4.1.5	The tool shall allow the selection of aircraft in all phases of flight (cruise, climb, or descent).	6.2.1
2.5.4.1.6	The tool shall also be capable of identifying aircraft as part of a spacing stream regardless of whether they are flying on or off published routes.	6.2.1
2.5.4.2	Identification of the spacing reference fix	
2.5.4.2.1	The Build-1 tool shall include the algorithms and graphical interface necessary to allow the manual selection of a 2-dimensional reference fix to which a spacing constraint can be applied.	3

2.5.4.2.2	It shall be possible for the user (controller) to select a spacing reference fix that lies either on or off a known/published position.	3.2
2.5.4.2.3	Arbitrary reference points shall be selectable by positioning a cursor over the desired position on the plan-view display or by entering a latitude and longitude for a point.	3.2.2
2.5.4.2.4	The GUI should provide an intuitive means by which to "capture" the cursor as it moves over the desired location.	3.2.2
2.5.4.2.5	The GUI shall also provide a means of defining a line-in-space.	3.2.2
2.5.4.2.6	Seagull shall explore the possibility of associating such a reference line with an arbitrary latitude/longitude, fixed arc, or a published sector or facility boundary.	3.2.2
2.5.4.2.7	Seagull shall explore the feasibility of these and other options for selecting a spacing reference fix and providing the appropriate graphical feedback to the user.	3.2
2.5.4.3	Trial planning for spacing and conflict resolution	
2.5.4.3.1	Build-1 tool shall provide a mechanism that allows the controller to manipulate "manually" individual trajectories, on a trial basis, in order to help generate advisories that solve for both spacing conformance and conflict resolution (trial planning).	7
2.5.4.3.2	Trial planning shall allow the controller to experiment with any combination of speed, altitude and heading changes in order to satisfy both conflict and spacing constraints.	7
2.5.4.3.3	The Build-1 tool shall provide the controller with an appropriate graphical interface for making heading/route changes for trial planning.	7
2.5.4.3.4	The trial-planning capability shall provide the controller with an efficient mechanism for making keyboard inputs of speed and altitude changes that minimizes heads-down time.	7

2.5.4.3.5	Flight-plan intent during trial planning shall be frequently updated with the latest controller inputs in order to provide continuous feedback of spacing and conflict status.	7
2.5.4.3.6	Once the controller accepts and activates the trial plan, the flight plan, as represented within CTAS, shall be amended to reflect the controller's intent.	7
2.5.4.3.7	To the extent possible, the Build-1 trial planning design should borrow from the basic "look and feel" of the trial planning implementation within the CPTP and D2 tools.	7
2.5.4.4	Graphical/textual feedback of spacing and conflict planning.	7
2.5.4.4.1	The Build-1 tool shall provide the controller with graphical and textual displays of predicted conflicts and spacing.	7 4
2.5.4.4.2	Predicted conflicts shall be indicated on the planview display and presented in a conflict list.	7
2.5.4.4.3	The conflict list shall provide information describing the aircraft pairs in conflict, predicted time to minimum separation, and predicted minimum separation distance.	7
2.5.4.4.4	The display of graphical and textual conflict information in the Build-1 design shall leverage heavily from the existing implementation within the CPTP and D2 CTAS tools.	7
2.5.4.4.5	The projected spacing of each aircraft in the spacing stream shall be presented on the planview display and shown numerically in a spacing list.	4 9.8
2.5.4.4.6	On the planview display, the aircraft positions shall be indicated by "spacing marker" which illustrates the spacing of each aircraft relative the first (next) aircraft that is predicted to cross, or be abeam to, the spacing reference fix.	9.8
2.5.4.4.7	The associated spacing list shall display each flight in the order of its arrival time abeam the reference fix, and provide spacing and arrival time information corresponding to the spacing markers on the planview display.	4

2.6	Sub Task 2 Requirements – Build 1 Software Implementation Create the software to implement the requirements and specifications documented in Sub Task 1.	9
2.6.1	Seagull shall develop and debug the Build-1 software in accordance with Sub Task 1 specifications and ensure compatibility with CTAS baselines processes.	9

APPENDIX B, User's Guide

B.1 Starting Miles-In-Trail Spacing Processing

1. Start CTAS in D2 mode:
   a. Run **cm** and attach to radar
   b. Run **pfs_c**
   c. Run four **ra**s
   d. Run **dp**
   e. Run **pgui** with the –upr –cp –alt and –default D2_DSR arguments on the command line.
2. On the PGUI's F1 panel, make sure that In Trail spacing is selected. (This is the default in the D2 mode.)
3. Press <Shift+F5> to bring up the Miles-In-Trail Spacing List panel. Subsequent selection of <Shift+F5> will toggle the panel (and the MSR panel, if it's visible) on and off but the spacing will continue to be calculated; CTAS considers the panel to be active in this case. The MIT panel may be positioned anywhere on the screen.
4. Begin processing by specifying a Spacing Reference, selecting the spacing distance and including aircraft in the Spacing Stream as described below.
5. As with all other CTAS windows, both the MIT and MSR panels are resizable, but neither will re-scale when stretched.
6. As with all other CTAS windows, neither the MIT nor the MSR panel may be dismissed by clicking on its window menu button (upper left frame corner), nor by selecting "Close" from that menu. Select <Shift+F5> to hide the panel(s), or *QUIT* to dismiss the panel(s) and end processing.

B.2 Selecting or Modifying a Spacing Reference

B.2.1 Defining a New Reference

There are two ways of defining a Spacing Reference:

1. Entering a named waypoint or runway value in the appropriate text field; or
2. Selecting a named waypoint or an arbitrary latitude/longitude location directly from the PGUI canvas.

A combination of methods 1 and 2 may be used to define a Line spacing reference.

Definition of the Spacing Reference on the PGUI canvas is accomplished using the middle mouse button.

If the operator elects to enter a named waypoint or runway in order to define a reference, the EDA software will verify the validity of the input before continuing. If the input is valid (a recognized named waypoint or runway) its text is capitalized and the Spacing Reference moves to the appropriate location on the PGUI. If the input is invalid, its text remains in lower case, an error message is displayed in the PGUI scratch pad, and no change is made to the Spacing Reference location (or, if it's a new reference, no spacing marker is displayed).

A reference can be defined in a few simple steps:

1. Select *New* from the *Spacing Reference* pull-down menu.
2. Click on the *Modify* button.
3. Select the type of reference by clicking on one of the radio buttons.
   a) POINT – Enter a named waypoint in the *Location* text field or click anywhere on the PGUI screen to specify the Point's location.
   b) LINE – Enter a known waypoint in the either or both the *Start* and *End* text fields, or mouse down anywhere on the PGUI screen and drag the cursor to define the line parameters. Release the mouse button to finish.

      Both the start and end positions may be edited by directly entering a named waypoint in the appropriate text field. The software allows for any combination of named waypoint/lat-lon pair to define a Line.
   c) ARC – Enter a named waypoint in the *Center Pt* text field or click anywhere on the PGUI screen to define the Arc's center point. Specify the Arc's radius by manipulating the *Radius* slider.
4. The default name for this reference will be displayed in the *Name* text field automatically as soon as a valid location is defined. To enter a different name, simply edit the *Name* text field.

   The naming convention for default names is based on the type of spacing reference selected and on the number of like references already defined. For instance, Point references have default names Point_*n*, where *n* is an integer ranging from 1 to MAX_ADVISORIES. At the time of this implementation, the value of MAX_ADVISORIES is 10.
5. Select the *DONE* button. The Modify Spacing Reference panel is dismissed and the newly defined reference name appears as the selected index in the Miles-In-Trail panel's *Spacing Reference* menu.

B.2.2 Selecting a Previously Defined Reference for Processing

All previously defined Spacing References are contained in the Miles-In-Trail panel's *Spacing Reference* pull-down list. If the Modify Spacing Reference panel is displayed, the desired reference may be selected from its *Spacing Reference* menu; either menu will summon the specified reference.

Select the desired Spacing Reference from either panel's *Spacing Reference* pull-down list and click on the *Select* button to begin processing with this reference.

B.2.3 Modifying a Previously Defined Reference

All previously defined Spacing References are contained in the Miles-In-Trail panel's *Spacing Reference* pull-down list. If the Modify Spacing Reference panel is displayed, the desired reference may be selected from its *Spacing Reference* menu; either menu will summon the specified reference.

1. Select the desired reference from the either panel's *Spacing Reference* pull-down list.

2. If the Modify Spacing Reference panel is not displayed, click on the *Modify* button to summon it.
3. Make the desired modifications to any or all of the MSR panel's text fields.
4. Select the *DONE* Button.
5. Select the *CANCEL* button at any time before hitting the *DONE* button to return to the original reference definition. The Modify Spacing Reference panel will be dismissed and the PGUI screen will display whatever Spacing Reference was previously defined.

B.2.4 Clearing a Spacing Reference

The Modify Spacing Reference panel's *CLEAR* button will erase any changes that have been made to the spacing reference fields, but will leave the panel displayed for further modifications.

B.2.5 Deleting a Spacing Reference

A Spacing Reference may be permanently removed from the *Spacing Reference* menu (both panels) by choosing the *DELETE* button on the Modify Spacing Reference panel when the target reference is displayed.

B.3 Selecting a Miles-In-Trail Spacing Distance

The in-trail distance is specified by manipulating the *Spacing* slider on the Spacing List window, or by entering a value in the slider's text field. The selected value is used to calculate the delta distances displayed on the Spacing List panel.

This distance can be changed at any time during processing to generate a new in-trail separation between aircraft.

B.4 Selecting/Deselecting Aircraft to be Included in Spacing Stream

Figure B-1 Including Flights in the Spacing Stream

Aircraft included in the Spacing Stream are differentiated from other aircraft on the screen by the presence of a salmon colored box around their position markers. Refer to Figure B-1 for a sample Spacing Stream.

To include a flight in the Spacing Stream, do the following:

1. Left click on an aircraft's position marker to bring up its flight data block.
2. Right click on the aircraft type (Row 3, Field 'A' of the flight data block) to add the aircraft to the Spacing Stream. A box will appear around the selected aircraft's position marker on the PGUI.
3. To remove a flight from the Spacing Stream, right click on Row 3, Field 'A' of its flight data block. The spacing reference box will no longer be shown around this flight's position marker.
4. A flight may also be removed from the Spacing Stream by left clicking directly on its position marker.

B.5 Displaying the Projected Spacing Stream

A special icon marks the projected position of each aircraft at the time of the lead aircraft's crossing of the Spacing Reference. These icons (sometimes referred to as "ghost" aircraft) may be displayed or dismissed by toggling the *Show Spacing* checkbox located at the bottom of the Miles-In-Trail Spacing panel as shown in Figure 2-1. Figure B-2 shows the spacing markers.

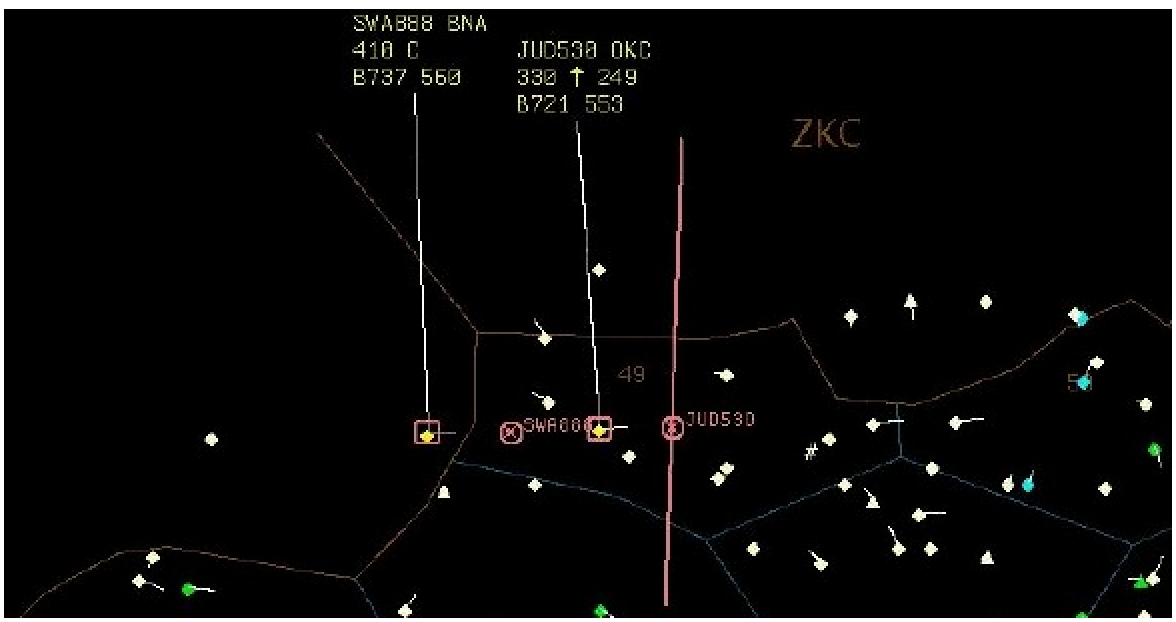


Figure B-2 Spacing Stream Position Markers

B.6 Selecting Aircraft to be Trial Planned

B.6.1 By Flight Path

1. Right click on the flight's destination in the first row of the flight data block.
2. When the new yellow trial plan path appears on PGUI, left click on the new path.

3. Using the middle mouse button, select the triangle within the circle icon and move to a new position.
4. Once the desired spacing is reached, choose the *ACCEPT* button in the Trial Plan window.

B.6.2 By Altitude

1. Right click on the flight's current altitude or flight level in the second row of its flight data block.
2. Select a new altitude from the list in the Trial Plan window.
3. Once the desired spacing is reached, choose the *ACCEPT* button in the Trial Plan window.

(Not available for arrivals at this time.)

B.6.2 By Speed

Speed trial planning is not implemented in the CTAS baseline at the time of this report.

B.7 Shutting Down Miles-In-Trail Spacing Processing

1. Selecting the *QUIT* button on the Spacing List panel will display a quit confirmation dialog box. Select *Confirm* to exit, *Cancel* to return to processing